AMIGA DOS

DAS MAGAZIN FÜR AKTIVE AMIGA-ANWENDER

DM 6,50
sfr 6,50
öS 52

3/90

AMIGA DOS

3/90
ISSN 0937-2717
DMV-Verlag

Hardware
Virenschutz am Floppyport

Messebericht CES
Die neuesten Amiga-Produkte

Viele Tips & Tricks
- Wochentag berechnen
- SpriteEditor Teil 2
- Tips zu Modula 2
- CLI und BASIC für Einsteiger

Listings
Super-Schachprogramm
Funktionsplotter in GFA-BASIC

Test
Mark II – Soundeditor
TurboPrint – das Druckerutility
The Publisher

MIT GROSSEM SPIELE-TEIL

STEFAN OSSOWSKI'S SCHATZTRUHE

(1) **Haushaltsbuch** bis zu 25 Konten, flexibel, leicht bedienbar, mit umfangreicher **deutschsprachiger** Dokumentation DM 8,-

(3) **MountainCAD** professionelles CAD-Programm, **deutsche Anleitung** DM 8,-

(4) **Spiele I, II, III** 10 erstklassige PD-Spiele aus allen Bereichen wie Action, Geschicklichkeit, Strategie (3 Disketten) DM 24,-

(5) **Anti-Virus** 8 Programme gegen alle Viren DM 8,-

(6) **Text** hochwertige **deutsche** Textverarbeitung DM 8,-

(7) **Utility-Disk** 25 nützliche Utilities aus allen Bereichen DM 8,-

(9) **Sonix-Paket** Original-Sonix-Player + 4 weitere Disketten mit phantastischer Sonix-Musik. **Top-Hit!** DM 40,-

(10) **Business** 3 Disketten: Tabellenkalkulation, relat. Datenbank, sehr gute Textverarb. Vers. engl. DM 24,- Vers. deutsch DM 70,-

(13) **Paranoid** sensationelles Breakout-Spiel DM 8,-

(14) **Buchhaltung** erstes **deutsches** PD-Buchhaltungsprogr. DM 8,-

(16) **AMIGA-Paint** sehr gutes **deutsches** Malprogramm DM 8,-

(17) **Videodatei** bringt Ordnung in Ihre Videodatei, **deutsch** DM 8,-

(18) **Fußballmanager** bei diesem Spiel können Sie Ihre Fähigkeiten als Manager eines Fußballclubs testen, **deutsch** DM 8,-

(19) **Platten/Cassetten/CD-Liste, deutsch** DM 8,-

(20) **Giroman** komfortables **deutsches** Programm, mit dem Sie Ihr Girokonto einfach verwalten können DM 8,-

(22) **Kampf um Eriador, V 2.0** taktisches Strategiespiel für 2 Personen mit sehr guter Grafik und Sound, **deutsch** DM 8,-

(25) **Label 2.0** Etikettendruckprogramm, **deutsch** DM 8,-

(26) **Risiko** die Amiga-Umsetz. d. bek. Brettspiels, **deutsch** DM 8,-

(27) **Ray-Tracing-Construction-Set, V 2.0** phantastisches Programm zur Berechnung von Licht und Schatten - siehe Test Amiga 1/88 - komplett auf **3** Disketten mit **deutscher Anleitung** DM 24,-

(28) **Wizard of Sound** ein phantastisches Musikprogramm zur Erstellung eigener Lieder, mit **deutscher** Anleitung (2 Disks) DM 10,-

(29) **Broker** ein sehr gutes **deutsches** Börsenspiel DM 8,-

(30) **Quickmenü** erst. Sie sich Ihre eig. Workbench i. **deutsch** DM 8,-

(31) **Blizzard** phantastisches Ballerspiel m. sehr guter Animat. DM 8,-

(32) **DSort deutsches** Diskettenkatalogisierungsprogramm DM 8,-

(33) **Pascal** ein komplettes Pascal-Paket (3 Disketten) mit Compiler, **deutscher Anleitung** u. einem s. gut. deutschen Editor DM 24,-

(34) **DiskKey** Diskettenmonitor mit **deutscher Anleitung** DM 8,-

(35) **Peters Quest** Geschicklichkeitsspiel mit lustiger Handlung und **deutscher Anleitung** DM 8,-

(36) **Spiele** auf dieser Diskette sind 3 Tetris-ähnl. Spiele enth. DM 8,-

(37) **MRBackup** Festplattensicherungsprogramm mit **deutscher Anleitung** DM 8,-

(38) **Universal-Datei deutsches** Datei-Verwaltungsprogr. DM 8,-

(39) **Assembler** ein komplettes Entwicklungssystem für Maschinensprache in **deutsch!** DM 8,-

(40) **Bibel-Quiz** lehrreich und unterhaltsam DM 8,-

(41) **Faktura** Fakturierungsprogramm inkl. Mahnungen und Adressen – mindestens 1 MB Speicher DM 8,-

(42) **BootMaster** Mit diesem Programm können Sie individuelle Bootblöcke mit Lauftext und Sternenhintergrund erstellen. DM 10,-

(43) **Banner II** Dieses Programm ermöglicht es Ihnen, komplette Banner mit Ihrem Drucker zu erstellen. Leicht bedienbar! DM 8,-

*** TAIFUN * TAIFUN * TAIFUN * TAIFUN * TAIFUN * TAIFUN ***

Die deutsche Public-Domain-Serie aus dem Hause Ossowski! Wußten Sie schon, daß beim PD-Versand Stefan Ossowski alle drei Monate 10 neue TAIFUN-Disketten mit den interessantesten Neuerscheinungen des PD-Marktes vorgestellt werden? Bevorzugt präsentieren wir Ihnen die neuesten deutschen Programme. Am 10. November erschienen die neuen TAIFUN-Disketten Nr. 111 bis 120.

Schnupperpreis: DM 53,- V-Scheck, DM 57,- Nachnahme

Versandkosten (Porto/Verpackung): Inland: DM 3,- V-Scheck DM 7,- Nachn. Ausland: DM 6,- V-Scheck DM 15,- Nachn.

Professionelle Anwendungssoftware für den anspruchsvollen User:

(100) **DME T & W = Text & Wörterbuch**
DME T & W ist ein leistungsstarker Texteditor sowohl für den Viel- als auch für den Gelegenheitsschreiber. DME T & W ist äußerst flexibel und sogar programmierbar. Sie erhalten DME T & W bei uns mit **deutscher Anleitung** und einem eingebauten **Deutsch-Englisch-Wörterbuch** mit ca. 3000 Vokabeln. DM 30,-
DME mit englischer Anleitung und ohne Wörterbuch: DM 6,-

(101) **RIM-5 = Relationale Datenbank**
RIM-5 ist eine äußerst leistungsfähige relationale Datenbank, die sowohl für den geschäftlichen als für den privaten Bereich prädestiniert ist. Natürlich liefern wir RIM-5 mit **deutscher Anleitung** und einem **ausführlichen Einführungskurs**. DM 30,-

(102) **AnalytiCalc = Tabellenkalkulation**
Eine leistungsstarke Tabellenkalkulation mit **deutscher Anleitung**. Schon nach kurzer Zeit wird AnalytiCalc ein unverzichtbares Hilfsmittel für Kalkulationsaufgaben im Büro oder zu Hause sein. DM 30,-

(103) **DEA Arithmetica = Die Göttin Arithmetica**
DEA Arithmetica ist ein Programm, auf das Schüler, Studenten und Lehrer schon lange gewartet haben. Ableitungen, Kurvendiskussionen und das Skizzieren von Funktionsgraphen stellen von nun an kein Problem mehr dar. **Made in Germany!** DM 30,-

(104) **Haushaltsbuch Version 2.0**
Die Komplettlösung zur Verwaltung Ihrer privaten Finanzen. Wollen Sie einen Überblick über Ihre Finanzen, Ihr Geld effektiver nutzen und Schwachstellen in Ihrer Budgetierung erkennen? Haushaltsbuch V. 2.0 ist leicht bedienbar und auch für Computerneulinge hervorragend geeignet! Demo-Dateien unterstützen sie bei Ihren ersten Schritten und das Programm ist frei von buchhalterischen Fachausdrücken. Ein Funktionsüberblick: Führen von verschiedenen Dateien, mehrere Kontenlisten, bis zu 10 Bilanzen, 40 freidefinierbare Konten, Suchroutinen, viele Voreinstellungen, doppelte Buchführung, Datei-Ex- und -Import, Filterfunktionen, Jahresübertrag, Mausunterstützung ... **Selbstverständlich mit umfangreicher deutscher Dokumentation!** Haushaltsbuch V 2.0 benötigt 1 MB Speicher. DM 98,-

(105) **Xytronic II**
Bei diesem Spiel können Sie intergalaktischen Handel betreiben. Um Ihren Reichtum zu mehren, müssen Sie geschickt kaufen und verkaufen und natürlich den Weltraumpiraten aus dem Weg gehen. Sehr gute Grafik und sehr guter Sound garantieren zusätzlich eine hohe Motivation. DM 29,-

(106) **Data-Manager**
Eine sehr leicht zu bedienende Dateiverwaltung, mit der Sie Adressen, Mitgliederlisten etc. perfekt verwalten können. Ein **deutsches Programm** mit **deutscher Anleitung**. DM 15,-

(107) **EGOS = European Game of Strategy**
Ein Spiel für 2 Spieler. Sie kämpfen um die Vorherrschaft in Europa. Versuchen Sie, die Taktik Ihres Gegners zu durchschauen und Ihre eigene Strategie entsprechend auszurichten. Ein Muß für Strategiefans. **Deutsch!** DM 29,-

(108) **Bundesliga 2000**
Ein leistungsstarkes Programm zur Bundesliga-Ergebnisverwaltung. Auf einer Diskette können Sie bis zu 40 Ligen verwalten. Mit Historie seit 1985. **Komplett in Deutsch!** DM 15,-

(109) **Money Player Deluxe – Geldspielgerät**
Sie fühlen sich wie in der Spielhalle! Start- und Risikoautomatik, Sonderspiele, viele Extras und ein toller Spielspaß. Spielkomfort durch Maussteuerung, Pal-Auflösung ... DM 39,-

(110) **Wizard of Sound 2.0**
WoS ist ein phantastisches Musik- und Komponierprogramm. Erstellen Sie eigene Lieder entweder durch Noten und per Klaviatur. **WoS** verfügt über 60 verschiedene Instrumente, einen Instant-Replay-Modus und eine große Funktionsvielfalt. **WoS** wurde **in Deutschland entwickelt** und wird mit **umfangreicher deutscher Dokumentation** komplett auf **2 Disketten** geliefert! Inkl. Demosongs und Player DM 35,-

(111) **Broker 2.0 + Das Börsenprogramm**
Broker 2.0 ist eine sehr **realistische Börsensimulation**. Alleine oder mit bis zu 3 Mitspielern versuchen sie Ihr Kapital zu mehren. Nur einer kann gewinnen! Dieses Spiel ist unterhaltsam, lehrreich, zeichnet sich durch lang anhaltende Motivation aus – 1MB Speicher DM 29,-

(112) **Vokabel-Trainer-English 1.0**
Der **Vokabel-Trainer** ist **ein einfach zu bedienendes, leistungsfähiges Lernprogramm**. Zusätzlich zu den bereits vorhandenen Vokabeln kann man den Wortschatz leicht erweitern. ein Programm, mit dem das Lernen Spaß macht! **Made in Germany!** DM 15,-

(113) **Pente**
Ein »**5-Gewinnt-Spiel**«, erweitert um einige Regeln. Sie spielen gegen den Computer. Für jung und alt mit hoher Motivation DM 19,-

Die Programme 100 bis 103, 105 bis 108 und 110 bis 113 erhalten Sie in dieser Ausführung ausschließlich beim PD-Vertrieb Stefan Ossowski!

ABO-SERVICE
Bei uns erhalten Sie fast jede PD-Serie auch im günstigen Abonnement! Auf unsere Staffelpreise gewähren wir außerdem einen **10%igen ABO-Rabatt**! Die Fish-Serie ist z. B. schon bis Nr. 260 lieferbar! Rufen Sie uns doch einfach an oder schreiben Sie uns, wenn Sie an weiteren Informationen zu unserem ABO-Service interessiert sind!

Zuverlässigkeit
\+ Schnelligkeit
\+ Service
= PD-Versand Stefan Ossowski

Testen Sie uns!

Wir führen alle bekannten PD-Serien wie Fish, Panorama, Faug Amicus, Auge, Taifun, Chiron, RPD, Kickstart, Slideshows, TBAG, Franz ... Viele Programme deutschsprachig.

5,-DM kosten unsere aktuellen **2** Katalogdisketten. Mit deutschem Inhaltsverzeichnis unseres gesamten PD-Angebots von weit über 1000 Disketten. Ab DM **4,50**. Gegen Verrechnungsscheck oder in Briefmarken anfordern.

Stefan Ossowski - Ihr PD-Spezialist - Veronikastr. 33, D-4300 Essen 1, Tel./Btx: 0201/788778

AMIGA im Aufwind

Mit dem Jahr 1990 ist eine neue Dekade der Technologie angebrochen. Diese Aussage trifft in besonderem Maße auch auf den AMIGA zu. Der Aufbruch ins professionelle Lager wurde ja bereits mit dem AMIGA 2500 eingeleitet. Doch erst sinnvolle Peripherie schafft das Umfeld zur professionellen Nutzung. So ist beispielsweise die DTP-Anwendung auf der Basis des AMIGA 2500 eine ernstzunehmende Alternative zu bestehenden DTP-Systemen, nicht nur in preislicher Hinsicht.

DTP ist aber sicherlich nur eine der professionellen Anwendungen, die ins Auge stechen. Denn unbemerkt hat der AMIGA auch in anderen Bereichen des täglichen Lebens Einzug gehalten. So mancher Trailer, der im Fernsehen jeden Abend zu sehen ist, findet seinen Ursprung in den enormen Grafikfähigkeiten des AMIGA. Bis vor wenigen Jahren waren Anwendungen dieser Art mit Heimcomputern nicht oder nur stark eingeschränkt möglich. Die Maschinen, die die gleiche Leistung erbrachten, wie es heute der AMIGA tut, waren für den Heimanwender unerschwinglich.

Mit den heute zur Verfügung stehenden Peripheriegeräten wie Genlock-Interfaces, Sound-Digitizern und Raytracing-Programmen erschließt sich auch dem Heimanwender, der nicht über unbegrenzte Geldmengen verfügt, die Welt der professionellen Video-Anwendung. Denn das einzige, was Sie neben Ihrem AMIGA benötigen, ist etwas Phantasie und das nötige Durchhaltevermögen, um ein angefangenes Projekt auch zu Ende zu bringen.

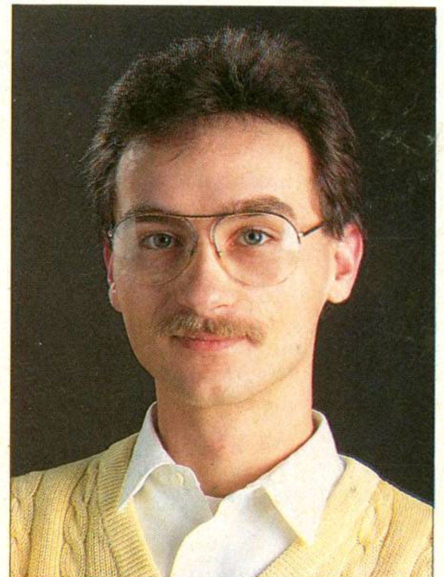
Wir wollen Ihnen bei all den Problemen, die bei der Arbeit auftauchen können, mit Rat und Tat zur Seite stehen. Manche Fragen werden sich bei genauer Durchsicht unserer Tips und Tricks sicherlich aufklären. Finden Sie in diesem Bereich nicht die Antworten auf Ihre Fragen: Hilfe ist nur einen Anruf bzw. Brief entfernt.

Bleibt uns nur noch, Ihnen viel Spaß bei der Lektüre der neuen AMIGA DOS zu wünschen.

Herzlichst Ihr

Markus Matejka

Markus Matejka

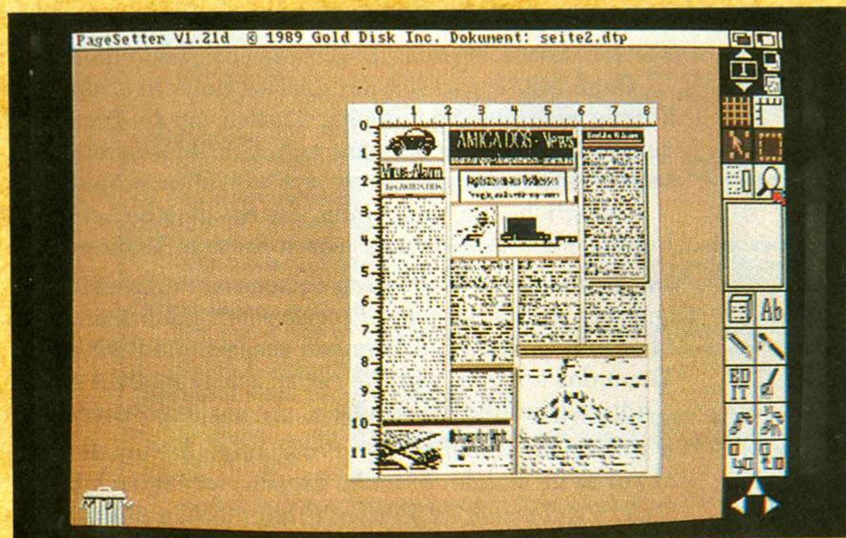


INHALT



Unser Programm des Monats: ein Schachprogramm, das sich nicht hinter kommerziellen Produkten zu verstecken braucht. Spielen Sie gegen den Computer, und gewinnen Sie, wenn möglich...

Seite 82



'The Publisher' erlaubt das Erstellen von kombinierten Text-/Grafikseiten. Ob das Programmpaket aus Textverarbeitung und Layout-Programm hält, was es verspricht, zeigen wir Ihnen auf...

Seite 26



Wer einen Drucker am Amiga angeschlossen hat, der hat bestimmt schon einmal Probleme beim Ausdruck gehabt. TurboPrint II heißt ein Utility, das jeden Drucker problemlos an den Amiga binden soll. Wir haben es für Sie getestet.

Seite 22

AMIGA DOS



Die CES in Las Vegas ist jedes Jahr Treffpunkt der Neuheiten. Wir haben uns für Sie auf dieser Messe umgesehen, über was es Neues zu berichten gibt. Lesen können Sie es ab

Seite 10



Falls wir Ihnen nicht bunt genug sind, schauen Sie doch mal bei Leisure Suit Larry II rein, denn Larry Laffer, Held der Story, treibt's diesmal noch bunter. Er ist wieder mal 'looking for love', aber, wie schon so oft, 'at several wrong places'...

Seite 123

LISTING

WERKSTATT

PUBLIC DOMAIN

SPIELE

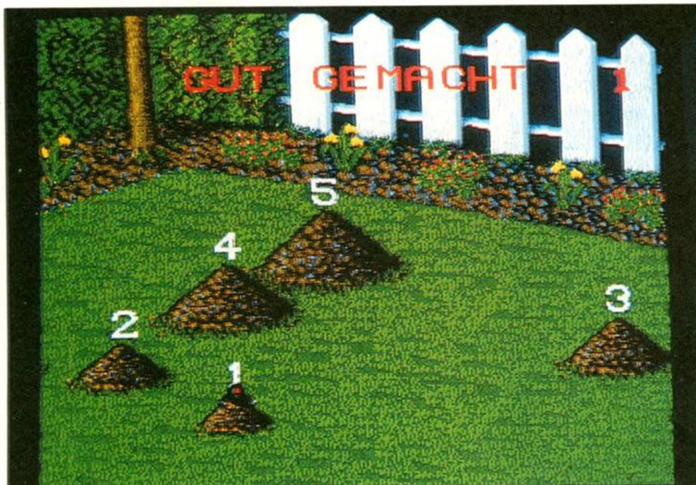
RUBRIKEN

Computerspiel – Computercomic!

Üblicherweise sind es Figuren aus Filmen und Comics, die von Softwareherstellern adaptiert werden und in Form eines Computerspiels auf den Verbraucher losgelassen werden. Das dies auch anders geht, haben nun Konami und Archie Comics bewiesen. Diese beiden Firmen haben beschlossen, aus der Telespielfigur Bayou Billy, einem aligatorwürgenden Glücksritter aus den weiten Sümpfen der Everglades, einen Comic-Helden zu machen. Von nun an werden Bayou Billys Abenteuer also als Telespiel und als Comic weitergeführt. Bisher gibt es das Ganze nur in den USA, ob wir hier in Deutschland einmal ein Bayou-Billy-Comic sehen werden oder ob das Spiel einmal für den Amiga vorliegen wird, ist sicherlich vom Erfolg abhängig.

Lernsoftware für Kinder bis acht Jahre

Wer auf der Suche nach Lernsoftware für Kinder ist, stellt recht bald fest, daß hier nicht gerade von einer reichen Auswahl die Rede sein kann. Mit 'Spielend Lernen' von Database Software gab es schon seit einiger Zeit einige passable Lernspiele für Kinder, die aber bislang nur in einer englischen Version erhältlich waren und somit für deutsche Kinder nicht nutzbar sind. Nun endlich liegen die Programme auch in einer deutschen Übersetzung vor. 'Spielend Lernen' gibt es für Kinder bis sechs Jahre und für Schulkinder von sechs bis acht Jahren. Ziel der Software ist es, spielerisches Lernen zu unterstützen. So enthält 'Spielend Lernen' für Kinder bis sechs Jahre grundlegende Schreib- und Rechenübungen, die in Spiele gekleidet wurden. Jede Version von 'Spielend Lernen' beinhaltet acht unterschiedliche Lernspiele. Bleibt für die Zukunft zu hoffen, daß dieser Vorstoß in Richtung Lernsoftware auf dem Amiga kein Einzelfall bleibt.



Spielend lernen mit dem Amiga

Computer gegen Raubkopierer

SAM – Security Application Microprozessorsystem – ist die Bezeichnung für ein universell einsetzbares Microcomputersystem. Dieses von der Firma ComProTec, Niederkassel, speziell für den Softwareschutz entwickelte System ist aufgrund seiner kompakten Bauweise in einem 60x50-mm-Standardgehäuse untergebracht und wird an der Parallelschnittstelle des Computers betrieben. Die Möglichkeiten von SAM reichen vom Auslagern programmspezifischer Informationen bis hin zur Auslagerung ganzer Programmteile, die dann von SAM abgearbeitet werden.

Innerhalb des Prozessorsystems ist ein Microcode-Chiffrierbaustein integriert, der die Codierung der durch SAM abzuarbeitenden Befehlscodes erlaubt. Dieser Baustein wird für jeden Kunden speziell programmiert, wodurch es möglich wird, die gesamte Dokumentation über SAM sowie alle über das System verfügbaren Informationen bekanntzugeben, ohne daß dabei die Sicherheit auch nur im geringsten beeinträchtigt würde. Da an dieser Stelle nicht alle Möglichkeiten, die das System bietet, aufgezählt werden können, hier eine kleine Auswahl der Möglichkeiten, die das System zu bieten hat:

- Installation von bis zu 64 verschiedenen Programmen/Programm-Modulen auf einem Adapter
- 128 Byte resetfester Datenspeicher
- Crack-Versuche werden vom Adapter protokolliert und können durch das Softwarehaus jederzeit ausgewertet werden.
- Reaktionen auf Crack-Versuche sind individuell steuerbar.
- Programmierung eines Dead-Counters für Demo-Versionen (Voll-Versionen, die sich lediglich n-mal starten lassen)
- Umprogrammierung eines Adapters zwecks Installierung weiterer Programme, Wandeln einer Demo-Version in eine Voll-Version etc. durch Software-Update beim Endanwender.

Neben dem Haupteinsatzgebiet von SAM – nämlich dem Softwareschutz – eignet sich SAM gleichfalls für den Einsatz im Bereich Datenschutz sowie für den Schutz von Hardware. Anwendungsbeispiele hierfür sind:

- Absicherung von Datenfernübertragungen
- Zugangsschutz zu Rechenanlagen
- Regelung von Zugriffsrechten
- Fileverschlüsselung (Text-Datenbankdateien etc.)
- SAM zum Schutz von Hardware vor Nachbauten und als Kommunikationsschnittstelle.

Die derzeit verfügbare Version kann an allen Amigas mit Parallel-Schnittstelle angeschlossen werden. Eine Version für serielle Schnittstellen sowie eine ON-Card-Version sind angekündigt.

Info:
ComProTec Hard- und Software Sicherheitssysteme GmbH, Langgasse 93
D-5216 Niederkassel 5

Microprose England, jetzt mit interner Programmierabteilung

Mit der Konfigurierung eines internen Programmierteams macht sich Microprose für die Anforderungen der 90er Jahre bereit. Um die dreiköpfige Kerntruppe, bestehend aus Tim Walther, Adrian Scotney und Malcolm Hellon, soll im Laufe des Jahres 1990 ein gößeres Team gebildet werden, dessen Aufgaben sowohl die Entwicklung eigener neuer Projekte als auch die Konvertierung von US-Software auf europaspezifische Computerformate sein werden. Paul Hibbard, Microproses Publishing Director, sagt dazu: "Die Formierung eines In-Haus-Programmierteams stellt für Microprose den Aufbruch in eine neue Ära des Wachstums dar. Innerhalb von drei Jahren gelang es Microprose, seine Marktposition vom Hersteller von Special Interest-Titeln zu einem der führenden Softwarehäuser auszubauen. Dieser Entwicklung Sorge tragend, war die Formierung eines In-Haus-Programmierteams ein natürlicher Schritt."

Computerferien – auch in den 90ern attraktiv

Ferien mit Bildung zu verbinden und eine ausgewogene Mischung aus Erholung und Information ist eine Urlaubsidee, die in den letzten Jahren immer mehr Freunde gefunden hat. Die Freiburger Firma Computer World bietet für

GFA für AMIGA

GFA-BASIC 3.5 Interpreter Amiga

Weiterentwicklung des GFA-BASIC 3.0 Interpreter mit 35 zusätzlichen Befehlen aus der linearen Algebra und Kombinatorik. Außerdem verbesserte Editor-Eigenschaften (Funktionen falten und Suche in Kopfzeilen gefalteter Funktionen bzw. Prozeduren) **DM 228,–** neu

GFA-BASIC 3.5 Compiler

Mit dem integrativen Compiler werden Ihre GFA-BASIC-Programme noch schneller.
Viele Optionen und Linker (kompatibel zu A-Link und B-Link) für andere Programmiersprachen im Lieferumfang enthalten. **DM 139,–** neu



Der Einstieg in GFA-BASIC 3.0 Amiga

Ein Lehrbuch für Programmieranfänger.
Dietmar Schell vermittelt auch dem unerfahrenen Programmierer Ideen und Anwendungsbeispiele für das Programmieren in GFA-BASIC. 248 Seiten, Hardcover, ISBN 3-89317-009-X **DM 29,–**



Training für Fortgeschrittene GFA-BASIC 3.0

Wer schon Erfahrung auf dem Amiga oder in irgendeinem BASIC-Dialekt hat, wird von den beiden Autoren bestens betreut.
Man erfährt und lernt eine Menge über Programmiertricks, nützliche und verwendbare Prozeduren, Anwendungen und die Besonderheiten des GFA-BASIC für Amiga. **DM 49,–**
329 Seiten, Hardcover, inkl. Diskette, ISBN 3-89317-010-3

neu

GFA-ASSEMBLER Amiga

Professioneller Makro-Assembler für 68000-Programmierer:
Leistungsfähiger Editor mit integriertem Assembler und Linker.
Nachladbarer Debugger.
Jetzt auch für die Commodore-Amiga-Computer lieferbar. **DM 149,–**

ZOETROPE

Das Computer-Animationssystem für Ihren Amiga mit der Funktionalität und den Eigenschaften, die man nur bei erstklassigen Grafiksystemen findet.
Das professionelle 2D-Animationsprogramm von ANTIC-Software, exklusiv von GFA. Umfangreiches Handbuch und Programm in Deutsch. **DM 198,–**

GFA-Gesamtkatalog anfordern
Anruf genügt
0211/5504-0

GFA Systemtechnik GmbH
Heerdter Sandberg 30-32
D-4000 Düsseldorf 11
Tel. 0211/5504-0 · Fax 0211/550444

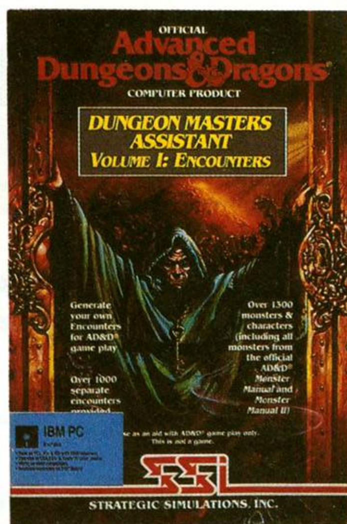


1990 wieder eine ganze Reihe von Computer Camps an, die neben Urlaub auch Kurse für die Teilnehmer bereithalten. Die Themen dieser Kurse reichen von BASIC über höhere Programmiersprachen wie Turbo Pascal bis zu solchen Spezialitäten wie Maschinensprache. Dafür, daß Kurzweil und Erholung in diesen Ferien nicht zu kurz kommen, sorgt ein sportliches Kontrastprogramm. Von Bumerang über Skateboard, Baseball und Football reicht hier die Bandbreite der Möglichkeiten.

Info: Computer World
Hurstweg 62b
7800 Freiburg

S.S.I. Rollenspieler-Werkzeuge nun auch für den Amiga

Dungeons & Dragons ist eines der beliebtesten papiergebundenen Rollenspiele der Welt. Die übermäßigen Würfel und Schreiborgien, die im Zuge eines solchen Spieles anfallen, lassen sich nun durch den Einsatz des Dungeon Master Assistant von S.S.I. wesentlich verkürzen. Dieses Programm nimmt den Spielern diese Aufgaben ab und erledigt sie im Handumdrehen. Neben diesem Programm ist auch ein neues Dragonlance Fantasy Rollenspiel angekündigt, Dragons of Lance. Auch dieses Programm wird es für den Amiga geben.



Neues von Hewson

Gleich einen ganzen Sack von Neuerscheinungen kündigt die Softwareschmiede aus England an. Insbesondere auf dem Amiga gab es aus diesem Hause bisher einige wirklich sehenswerte Spiele. So ist es dann auch nicht verwunderlich, daß der Reigen der Neuerscheinungen vom zweiten Teil des Spielehits Nebulus eröffnet wird. Auch der eben gerade für den Amiga erschienene Titel Stormlord erhält eine Fortsetzung. Das mehrfach ausgezeichnete Roboterspiel Paradroid gibt es endlich auch in einer Amiga-Version. Die C-64-Version von Paradroid war eines der attraktivsten und spielbarsten Programme aus 1988.

Da es mit Konvertierungen und Neuauflagen alter Software nicht getan ist, werden mit Scavenger, Mazemania und 5th Gear auch drei neue Originaltitel vorgestellt.

Während es sich bei den ersten beiden Titeln um Arcaden-Abenteuer mit Jump-&-Run- bzw. Labyrinth-Einschlag handelt, ist 5th Gear eine neue Rennsimulation.

Professional Tools – Betriebssystemerweiterung für den Amiga

UNIX ist das große Vorbild, an dem sich Professional Tools orientiert. Mit Professional Tools liegt nun eine Betriebssystemerweiterung vor, die dem Anwender über 100 neue Befehle zur Verfügung stellt, die die bisherigen Befehle des Amiga-Betriebssystems erweitern. Die neu konzipierte Umgebungsshell ermöglicht nun sechs verschiedene Arten der Aus- und Eingabeumlenkung, Funktionstastenbelegung, Protokoll-Option sowie die Verwendung von numerischen- wie auch Stringvariablen. Die Verwendung echter Variablen und die Möglichkeit der Schleifenprogrammierung (DO-UNTIL, DO-WHILE, DOWN-REPEAT) in Batch-Dateien sowie die Verwendung von Unterprogrammen (SUB, CALL usw.) erhöhen die Leistungsfähigkeit des DOS gewaltig. Weiteres hervorstechendes Feature ist ein neuartiger Handler, der es ermöglicht (wie bei großen Desktop-Workstations), die Textausgabe eines CLI-Kommandos an die Benutzeroberfläche INTUITION zu senden, damit der Anwender sie dort nachbearbeiten kann. Der Text läßt sich beliebig betrachten, formatieren oder grafisch aufbereiten – ähnlich wie mit einem Desktop-Publishing-Programm. Des weiteren findet der Anwender zusätzliche Wildcards, einen Linker, Timerkommandos, Diskmonitorbefehle sowie eine Vielzahl von Dateibefehlen wie zum Beispiel zum Komprimieren von ASCII-Dateien oder zum Drucker-Spooling.

Info:
GSS, Giero System Software, Sven Giero, Lessingstraße 14, 3167 Burgdorf



Amiga Fraktal Generator 3 D



Gibt es wieder Gratis-Chips bei DMV?
HANNOVER MESSE CeBIT '90
21. - 28. MÄRZ 1990
Wir stellen aus:
Halle 7
Stand E50



Supergrafik im Sekundentakt

Vergessen Sie alles, was Sie bisher über Fraktalgrafik-Programme gehört haben – die unendliche Weite phantastischer Bilder erschließt sich nur über ein superschnelles Programm: **Fraktal Generator 3D**

High-Speed
Nur noch 7 Sekunden für das Urbild!

Super-Parallel-Projektion
Frei wählbarer horizontaler Blickwinkel mit 360 Grad:
Betrachten Sie das "Fraktalobjekt" von allen Seiten
Stufenloser vertikaler Blickwinkel:
Wahlweise Sicht von oben und unten, schräg oder in der Totalen

Speicherung im IFF-Standard
Einladen der Fraktal-Bilder in Mal- und Zeichenprogramme:
Verwendung als Hintergrund, Motiv oder Vorlage

Voller Bedienungskomfort
Auswahl komplett mit Pull-Down-Menüs
Wahlweise Maus- oder Tastensteuerung

Phantastische Farbmöglichkeiten
32 Farben im Low-Resolution-PAL-Modus
Eigenes Farbrequester mit stufenloser Schiebereglung

Mehrere separate Bildspeicher
Bis zu vier Bilder gleichzeitig abrufbar
Separate Farbzuordnung und Animationsmöglichkeit

Farb-Animationen
Phantastische Effekte durch Amiga-Color-Cycling

Amiga Fraktal Generator 3 D

3 1/2"-Disk. Best.-Nr. 2901

69,- DM (unverbindliche Preisempfehlung)
Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

Inland:		**Ausland:**	
Einzelpreis	69,– DM	Einzelpreis	69,– DM
zzgl. Versandkosten	4,– DM	zzgl. Versandkosten	6,– DM
Endpreis	**73,– DM**	**Endpreis**	**75,– DM**

– Bitte benutzen Sie die Bestellkarte im Heft. –

DMV-Verlag · Postfach 250 · 3440 Eschwege
DMV Daten- und Medienverlag



SHOWDOWN

in LAS VEGAS

The show must go on

Jedes Jahr im Januar steht Las Vegas völlig im Zeichen der Consumer Electronic Show. Hightech und Entertainment sind die Themen, die Aussteller, Händler und Journalisten aus aller Welt in das Spielerparadies Amerikas mitten in der Wüste locken. Hier besteht die Möglichkeit, die neuesten Produkte, ob Hardware oder Software, in Augenschein zu nehmen. Dabei geht es nicht nur um Computer, Software oder Videospiele. Das Angebot reicht von Videokameras und Stereoanlagen bis hin zu Alarmsystemen, Bildplattenspielern, Holographieanlagen, Telefontechnik und vielem mehr. Zu finden ist die CONSUMER ELECTRONIC SHOW im Convention Center, gerade einen Steinwurf entfernt vom berühmten Las Vegas Strip mit all seinen Casinos. AMIGA DOS hat sich dort einmal für Sie umgesehen, denn es lohnt sich wirklich, die dort zur Schau gestellten Produkte einmal genau zu betrachten.

Amiga News

Die CES, die in einem halbjährlichen Turnus stattfindet, hatte auch diesmal mit einigen Überraschungen aufzuwarten. Der Amiga ist zwar prozentual nicht so stark verbreitet wie in Deutschland, an absoluten Stückzahlen gemessen jedoch weit mehr vertreten als in Deutschland.

Commodore selbst war zwar nur mit einem zirka 20 qm umfassenden Stand vertreten, dafür brachen hauptsächlich die Softwarehersteller mehr als eine Lanze für dieses System. Entsprechend dem Umfeld gab es viel neue Entertainmentsoftware zu sehen, die so manches Amiganerherz höher schlagen lassen wird. Zwar muß damit gerechnet werden, daß noch mindestens ein bis zwei Monate vergehen, bis die überwiegende Anzahl der Titel auch in Deutschland erhältlich sein wird, das Warten darauf lohnt sich allerdings in jedem Fall. Der Trend geht deutlich in Richtung immer anspruchsvollerer Software.

Reine Actionspiele sind weitgehend out, der Trend geht deutlich in Richtung komplexe Spiele. Eindeutige Spitzenreiter sind die Adventures, Rollenspiele und Storyadventures, die dem Spieler so manche harte Nuß zu knacken geben. Prognostizierte Spielzeiten von 200 bis 400 Stunden sind in diesem Zusammenhang keine Seltenheit. Dies bedeutet auf der anderen Seite, daß die Zeiten (hoffentlich) vorbei sind, wo man viel Geld für ein Full-Price-Produkt ausgab und es bereits nach wenigen Stunden entnervt wieder in die Schublade legte, weil entweder Grafik oder Spielwitz allzusehr zu wünschen übrig ließen bzw. sich das Spiel als schlicht und ergreifend unspielbar präsentierte.

Schlechte Zeiten auch für Raubkopierer, da verstärkt Anstrengungen unternommen wurden, diesem Mißstand einen Riegel vorzuschieben. Ob Dongle, 150seitige Manuals oder spezielles Präparieren der Diskette, das unerlaubte Kopieren von Software wird zumindest sehr erschwert.

Eine interessante Problemlösung verspricht die Firma CDS, die auf der Messe ihre drahtlose Maus für den Amiga vorstellte. Somit entfällt das lästige Kabel, das schon so manche Kaffetasse umgerissen hat und den Benutzer damit fast zur Verzweiflung brachte. Die Maus kann drahtlos bis zu einer Entfernung von 10 Metern betrieben werden und wird für zirka 200 DM bald auch in Deutschland erhältlich sein. Aber damit nicht genug, wurde doch bei der Konstruktion der Maus ein weiteres Übel an der Wurzel gepackt. Selbst Computermäuse brauchen einen gewissen Auslauf auf dem Schreibtisch, und nur allzuoft ist dieser Platz einfach nicht vorhanden. Abhilfe schafft hier ein kleiner, seitlich angebrachter Taster, der im gedrückten Zustand den benötigten Auslauf auf 10 % der sonst benötigten Fläche reduziert. Eine sehr praktische und nützliche Sache, zumal auch der Preis attraktiv ist.

Holographie wie noch nie

Sehr interessant gestaltete sich die Vorführung eines neuen Holographiesystems, das im Gegensatz zu anderen Systemen nicht nur wesentliche klarere Bilder zeigt, sondern auch in der Lage ist, kleine Filme ablaufen zu lassen. Das Gerät an sich ist von Form und Größe einer kleinen Litfaßsäule nicht unähnlich und hat im mittleren Bereich eine Aussparung, in der die Hologramme betrachtet werden können. Tatsächlich kann man in diese Ausparung auch hineinfassen, eine Versuchung, der nur wenige Besucher dieses Standes widerstehen konnten. Denn es war schon ein wenig verwunderlich, tropische Zierfische in der Luft herumschwimmen zu sehen. Die Qualität der Darstellung war dabei so gut, daß die Fische per Augenschein kaum von ihren Artgenossen im Aquarium zu unterscheiden waren. Wenig später war ein zirka halbstündiger Film zu bewundern, in dem Ausschnitte aus "Schwanensee" zu sehen waren. Die ungefähr 10 cm große Ballerina sowie die Bühnendekoration erzeugten dabei einen so plastischen Eindruck, daß man ständig der Versuchung erlag, sie aus der Grundplatte herauszunehmen und auf der Hand weitertanzen zu lassen. Laut Herstellerangaben wird das System zirka 15.000 US$ kosten, mit Hilfe des etwa genauso teuren Aufzeichnungsgerätes können dann Filme für dieses System selbst erstellt werden.

VideoMania

Waren es vor Zweijahresfrist noch die diversen Computertypen, die Begeisterung auslösten, so sind es heute unzweifelhaft die Videospiele, die in einer nie dagewesenen Vielfalt auf den amerikanischen Markt drängen. Zieht man ins Kalkül, daß der europäische Markt diesem Trend mit einer gewissen Verzögerung zu folgen pflegt, so steht uns eine wahre Videospielschwemme ins Haus. Anzeichen dafür gibt es bereits genug, so konnte beispielsweise das Nintendo Entertainment System immerhin 180000mal in Deutschland verkauft werden. Doch ist dies beileibe nicht das einzige Videospielsystem, das bereits eine Fangemeinde erobern konnte.

Als Alternative beziehungsweise Ergänzung stehen dem Videospielfan das Sega Master System und die PC-Engine zur Verfügung. Diesen 8-Bit-Geräten steht eine wachsende Anzahl an 16-Bit Geräten gegenüber, deren bekanntester Verteter zweifellos die 16-Bit-Konsole von Sega ist. Dies ist um so bedeutsamer, da die Grafik- und Soundfähigkeiten dieser Maschine, speziell im Zusammenhang mit bewegten Grafiken, schon dicht an die Möglichkeiten des Amiga heranreichen. Echte Konkurrenz bekommt dieses System zweifellos mit der neuen Spielkonsole von NEC, die mit einem Laserdisk-Player aufwarten kann. (Technisch ist das Turbo-Grafx-System von NEC mit der PC-Engine fast identisch.)



Bild 1: Das Sega Genesis System mit Monitor, Modem und Softwaretiteln



Bild 2: Das Turbo-Grafx-System von NEC mit CD-ROM

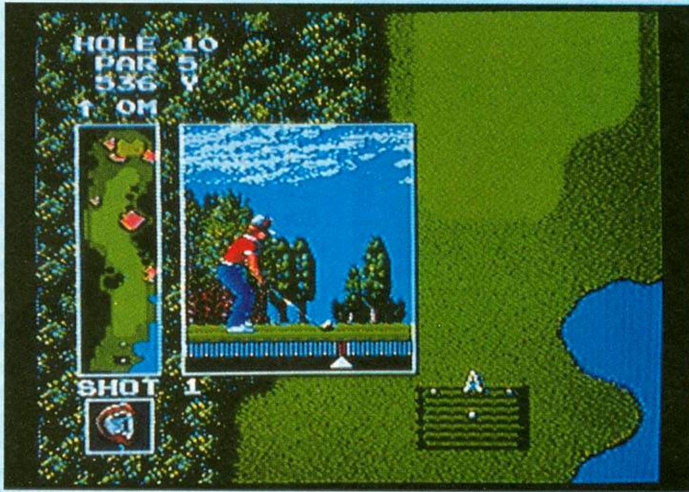


Bild 3: Screenshot Power-Golf für das Turbo-Grafx-System von NEC

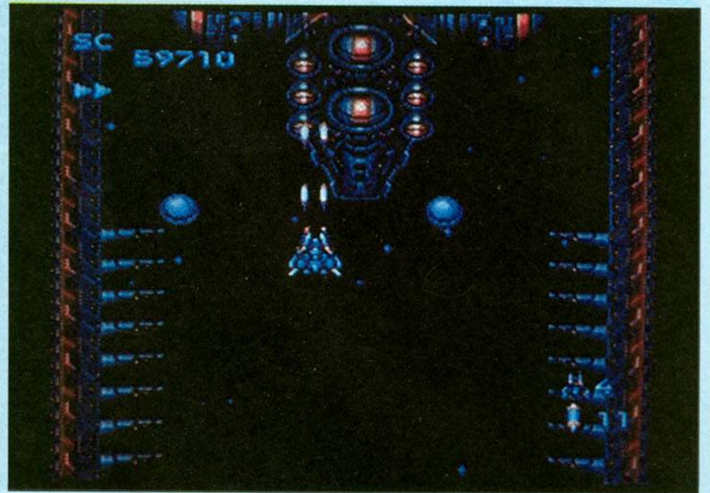


Bild 4: Screenshot Blazing Lazers für das Turbo-Grafx-System von NEC

Mit Hilfe dieser Laserdisk stehen natürlich ganz neue Möglichkeiten in bezug auf Realtime-Grafiken zur Verfügung, als dies mit den bisher verwendeteten Cartridges möglich war. Echte Walk-through- bzw. Fly-through-Szenarien sind nun möglich, die einen bisher ungeahnten Spielspaß an einer Videokonsole ermöglichen. Denn mit einem Speicherplatz von zirka 550 MByte können jetzt Spiele realisiert werden, mit deren Grafikumfang selbst eine Harddisk überfordert wäre. Der neueste Clou: Cinemaware präsentierte ihre Videospielversion von "It came from the desert" auf dem Turbo-Grafx-System von NEC. Während schon die Amiga-Version ein echter Hit war, nicht nur wegen der sehr gut gelungenen Grafiken, verspricht die Videospielversion auf CD noch viel mehr Spaß. Denn mit dieser Version wird nun erstmals bewegte, digitalisierte Grafik mit Makroaufnahmen von Ameisen so kombiniert, daß man sich unwillkürlich in Filme wie Formicula versetzt fühlt. Doch realistische Grafik (in Realtime!) reichte den Machern von Cinemaware nicht aus. Vorbei sind die Zeiten, wo klägliche Sound-FX jeden Realismus im Keim erstickten. Sound und jede Art von Geräuscheffekten werden ebenso wie die Grafiken digitalisiert und runden den realistischen Gesamteindruck ab. Eine ungefähre Vorstellung von den grafischen Fähigkeiten vermittelt Ihnen das Bild unseres USA-Korrespondenten Marshal Rosenthal, der sich selbst durch die drohende Invasion der Riesenameisen nicht davon abhalten ließ, mit einer der Hauptdarstellerinnen zu flirten. Laut Herstellerangaben werden bis zum Jahresende zirka 20 Spieltitel dieser Art auf CD vorliegen. Während die Gamecards des Turbo-Grafx-Systems auch problemlos auf der PC-Engine laufen (und umgekehrt), ist die Frage der Kompatibilität der CD-Software letztendlich noch nicht geklärt.



Bild 5: Screenshot Victory Run für das Turbo-Grafx-System von NEC



Bild 6: Derart realistische Grafiken wie hier bei Probeaufnahmen für CD-Software für das Turbo-Grafx-System gibt es sonst bei keiner Videokonsole

Als Beigabe ist das CD-System auch als handelsüblicher CD-Player verwendbar und ist auch in der Lage, die neue CD-Generation zu verarbeiten, die neben der Musik noch Bilder, beispielsweise der Popgruppe, auf dem Fernseher darstellen kann. Der Preis für das Turbo-Grafx System beträgt zirka 199 $. Das CD-Laufwerk liegt preislich durchaus im Rahmen anderer, handelsüblicher CD-Player.

Zumindest für diejenigen, die ihren Computer ausschließlich zum Spielen verwenden, sind diese Konsolen ebenso wie die dafür angebotene Software eine echte Alternative.

Ergänzt wird diese Produktlinie durch mehrere portable Systeme, deren bekanntesten Vertreter der Gameboy von NINTENDO sowie das Lynx-System von Atari sind. Das Lynx-System wurde ursprünglich von EPYX entwickelt und bietet im Gegensatz zum Gameboy einen farbigen LCD-Screen mit 4096 Farben. Leider stehen für dieses System erst fünf Spiele zur Verfügung, wahrend der mit einem Monochrom-Bildschirm ausgestattete Gameboy bereits über eine große Anzahl an Spieletiteln verfügt. So war es auch keine große Überraschung, daß eine Vielzahl der Titel, die bereits für das Nintendo Entertainment System existieren, nun nach und nach auch für den Gameboy umgesetzt werden. Leider eben mit der Einschränkung, daß nur monochrom gespielt werden kann, was sich besonders bei Spielen, die den Anwender grafisch ansprechen sollen, bemerkbar macht.

(mm)

Ein weiterer Anziehungspunkt waren Neuerungen im Bereich Autoalarmanlagen. Invisibeam stellt da eine relativ neue Lösungsmöglichkeit dar. Um das Auto herum wird ein unsichtbares Feld erzeugt. Nähert sich nun ein Mensch bis auf eine geringe Distanz dem Fahrzeug, so wird er von der Alarmanlage aufgefordert, den Sicherheitsbereich des Autos zu verlassen. Entfernt sich die Person nicht, wird nach wenigen Sekunden ein akustischer Countdown ausgelöst. Wenn die betreffende Person bis dahin nicht das Weite gesucht hat, wird ein Alarm ausgelöst, der die Hupe, Licht, Blinker etc. aktiviert.

Schlechte Zeiten für Autodiebe

Was aber nun, wenn der rechtmäßige Besitzer sich seinem so gesicherten Auto nähert?
Ganz einfach: Der Besitzer braucht bloß ein Codewort auszusprechen und kann damit die Anlage deaktivieren. Der Clou ist, daß die Alarmanlage eine Stimmanalyse durchführt und somit den Besitzer von allen anderen Personen unterscheiden kann. Es funktioniert tatsächlich, wir haben uns vor Ort davon überzeugt. Natürlich kann das Gerät ebenfalls mit einem Codewort des Besitzers wieder aktiviert werden.
Interessierte Besitzer von Edelkarossen können Invisibeam bei folgender Adresse beziehen:

International Distributors
CANADA: M.V.P.
Marketing Inc. 4407
Garand Street, St. Laurent,
Quebec H4R 1X2
(514) 336-0462

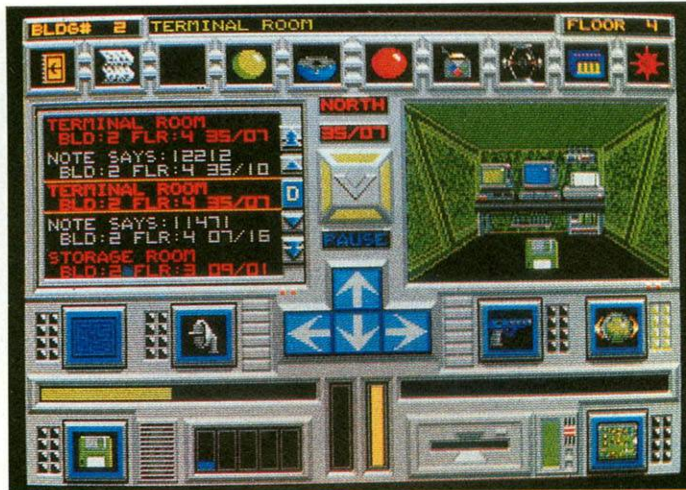
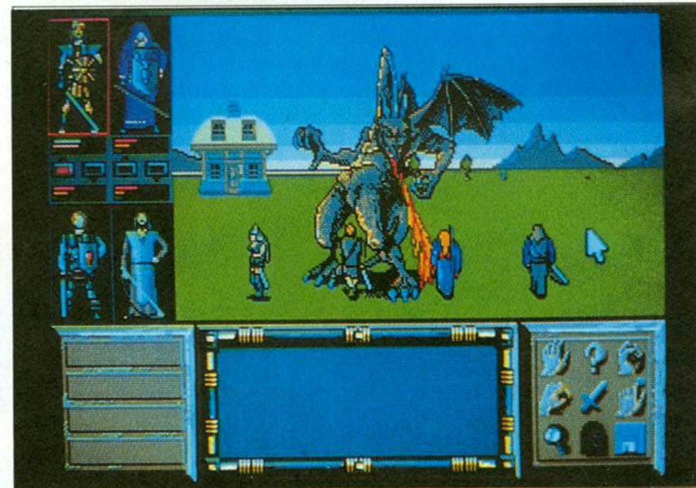


Bild 1: Screenshot Day of the Viper

Games, Games,

Wenn man von Computerspielen spricht, dürfen namhafte Firmen wie Origin, Mediagenic, Lucasfilm und Cinemaware nicht fehlen. Sie und viele andere gaben sich ein Stelldichein, um die letzten Neuheiten ihrer Game-Designer zu präsentieren. Und es ist wahrhaft erstaunlich, was in nächster Zeit auf den spielfreudigen Amiganer einstürmen wird. Dabei gibt es nur ein Problem: Der Trend geht hin zu immer besseren, farbenfroheren und komplexeren Abenteuern, so daß es allmählich schwierig wird, dies alles zu bewältigen. Denn kaum jemand kann mehrere hundert Stunden für ein und dasselbe Spiel aufbringen (es sei denn im Urlaub), wenn es einen ganzen Berg neuer Software gibt. Leider gab es auch einige weniger erfreuliche Randnotizen. Epyx, die durch ihre sehr gut gelungenen Sportumsetzungen bekannt sind, sind leider ab 1990 “out of business“. Ursache dafür war das Engagement für das Atari Lynx System, ein portables Telespiel von der Größe einer Videokassette. Das System selbst ist zwar beeindruckend in seiner Farbenpracht und Aufmachung, auch die Spiele, die für dieses System angeboten werden, sind ausgezeichnet, doch die Kosten für Entwicklung und Fertigung bis zur Serienreife waren immens. Bleibt zu hoffen, daß der Versuch der Konsolidierung erfolgreich ist und EPYX in naher Zukunft wieder auf dem Markt vertreten ist. Überhaupt kann man sagen, daß in den Vereinigten Staaten eine wahre Videomanie ausgebrochen ist. Nähere Informationen zu diesem Thema und zu dem neuen Videospielsystem von NEC finden Sie in unter der Headline “VIDEOMANIA“.


Bild 2: Screenshot Drakkhen

Wer allerdings einen Commodore-Stand nach deutschem Vorbild erwartet hatte, der wurde ein wenig enttäuscht. Ungefähr 20 qm hatte Commodore Amerika zur Präsentation ihrer Neuheiten zur Verfügung. Leider gab es dort nur wenig Neues zu bewundern, zumal Kernstück des Standes ein Verkaufspaket mit dem C-64 und einem Flugsimulator war. Neben diesem Komplettpaket waren noch ein Vertreter der neuen PC-Linie und ein Amiga 500 zu sehen.



Bild 3: Screenshot Champions of Krynn

Rollenspiel von Accolade

Eine sehr angenehme Neuheit von Accolade steht ins Haus. Accolade konnte ja besonders mit ihren Simulationen wie Testdrive eine große Fangemeinde erobern. Um so erfreulicher ist die Tatsache, daß in Zukunft auch der Adventure- und Rollenspielmarkt von Accolade bedient wird. “Day of the Viper“, so der erste Titel in diesem Genre, ist ein Weltraum-Action-Adventure, in dem Sie einen Droiden mittels Fernsteuerung durch eine Raumstation lenken. Ziel ist es, die Verteidigungssysteme der Raumstation zu reaktivieren, um den Invasionsplänen eines außer Kontrolle geratenen Androiden ein Ende zu machen.

Sollte dies nicht gelingen, so hat die Menschheit wenig Chancen, dem Angriff des Androiden zu entgehen. Neu ist vor allem, daß Kämpfe nicht im bekannten Zug-um-Zug-Verfahren durchgeführt werden, sondern in Realtime. Da kann es durchaus einmal passieren, daß Sie von vier verschiedenen Gegnern aus vier Richtungen gleichzeitig angegriffen werden. Hier können dann nur noch schnelle Reaktionen ein drohendes Desaster verhindern. Zirka 30 verschiedene Roboter und Droidentypen gilt es dabei zu besiegen, und jeder Typ hat seine eigenen Stärken und Schwächen, die Sie im Laufe des Abenteuers selbst herausfinden müssen.



Bild 4: Der Commodore Stand auf der C.E.S.

Drakkhen

Normalerweise sind, zumindest in Adventurespielen, die Drachen den in der Spielwelt lebenden Charakteren nicht sonderlich wohlgesonnen. So verwundert auch die Vorgeschichte von Drakkhen den versierten Computerspieler kaum. Ein Kämpfer dringt in eine verborgene Höhle ein und findet dort einen Drachen. Kurzerhand wird dieser Drache gemeuchelt, was allerdings tiefergehende Konsequenzen hat, wie der "tapfere" Kämpfer zunächst nicht bemerkt. Denn als Folge des Todes des Drachens kommt sämtliche Magie zum Stillstand.
Szenenwechsel. Im Hafen der Stadt legt ein Schiff an, dem ein Windmagier entsteigt. Dieser klärt nun die Bewohner darüber auf, daß der Tod des letzten Drachen für den Ausfall jeglicher Magie verantwortlich ist. Einzige Lösung: Eine tapfere und findige Party muß sich auf den Weg machen, um etwas zu finden, was dem Drachen neues Leben einhaucht. Ein faszinierendes Abenteuer, das durch seine ungewöhnliche Storyline ebenso besticht wie durch tolle Grafiken und ein Walk-through-Szenario. Mit einer prognostizierten Spielzeit von zirka 200 bis 400 Stunden wird hier wirklich Value for Money geboten.

S.S.I. – Der D-&-D-Spezialist setzt die erfolgreiche Serie fort

Ausgehend von der Welt, die in DRAGONLANCE bereits beschrieben wurde, können nun neue Abenteuer bewältigt werden. Champions of Krynn, so der neue Titel aus dem Hause S.S.I., kann jedoch mit einigen Neuerungen aufwarten, die vor allem das problematische Handling der Charaktere betreffen. Hier wurde nun das probate System aus Spielen wie Pool of Radiance und Curse of the Azure Bonds übernommen. Diesmal gilt es, die Dunkle Königin zu bekämpfen, die unrechtmäßigerweise den Thron für sich in Anspruch genommen hat. Natürlich können auch in Champions of Krynn viele Charaktere wiedergetroffen werden, wie zum Beispiel Tanis und Tasslehoff. Neuheiten gibt es auch im Magiesystem. Der Grafikscreen enthält nun eine Anzeige für den aktuellen Mondstand, der direkte Auswirkungen auf die magischen Fähigkeiten der Zauberer hat. Götter zeigen sich in diesem Spiel von der großzügigen Seite, und Priester können mit außergwöhnlichen Fähigkeiten beschenkt werden. Champions of Krynn wird für den Amiga im Frühsommer erhältlich sein. Auch mit neuen Monstern wird nicht gespart. Neben Draconiern gibt es neue Drachen und einen Death Knight, der den Abenteurern erhebliche Schwierigkeiten bereiten wird. Auch von der grafischen Seite wurde das Abenteuer gehörig aufpoliert.



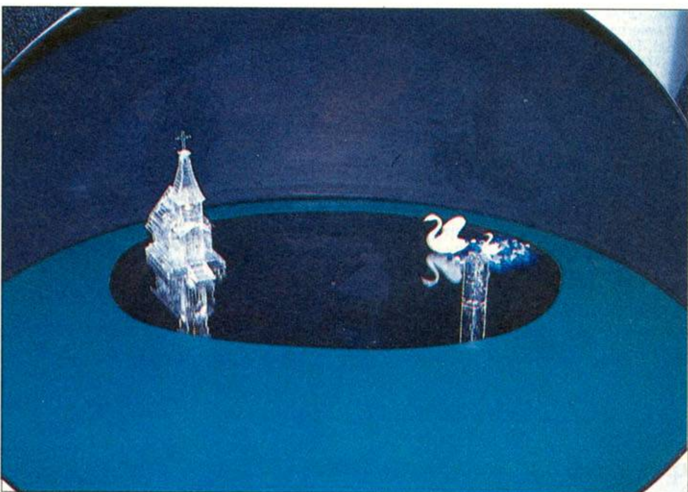

Bild 5: Ein neues Holographiesystem wußte durch verblüffenden Realismus zu überzeugen


Bild 6 und 7: Auch viele Superhelden der Comicszene hatten sich auf der C.E.S. ein Stelldichein gegeben



Bild 8: Screenshot Unreal

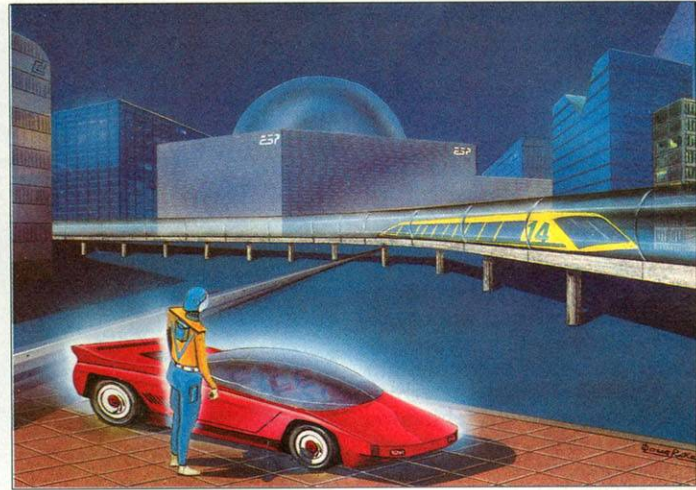


Bild 9: Die neue Alarmanlage von Invisibeam

The Champions of Krynn wird in Deutschland im Frühsommer erhältlich sein.

UBI-Softs Unreal

Auch die französischen Softwarehersteller ließen es sich nicht nehmen, auf der C.E.S. ihre neuesten Produkte zu zeigen. Am meisten beeindruckte uns in diesem Zusammenhang "Unreal", ein Actionadventure mit einer für dieses Genre faszinierenden Grafik. Unreal besteht dabei prinzipiell aus zwei Teilen: zum einen ein horizontal scrollendes Szenario, in dem sich der Held dieses Actionadventures tapfer mit seinem Schwert gegen die reichlich vorhandenen Feinde wehrt, zum anderen eine Passage, in der unser Held jeweils zum nächsten Landstrich gelangt.

Leider lag uns bei Redaktionsschluß noch keine eindeutige Aussage vor, wann Unreal in Deutschland erhältlich sein wird.

Natürlich können wir Ihnen im Rahmen dieses Berichtes nicht sämtliche Neuerscheinungen vorstellen, aber zumindest einen Querschnitt durch die Reihe der Softwaretitel geben. Wir werden Sie auch in Zukunft über die Entwicklung auf dem amerikanischen Markt auf dem laufenden halten, der ja starke Auswirkungen auch auf die europäische Angebotspalette hat. Die nächste Consumer Electronic Show findet übrigens Anfang Juni in Chikago statt.

(mm)





CPS Computertechnik GmbH
Hamburger Str. 283 · 3300 Braunschweig
Fax (05 31) 33 06 61

Ladenöffnungszeiten: Mo-Fr 9-18 Uhr
Sa 9-13 Uhr
Studentenrabatte geg. Nachweis a. Anfrage!

 (05 31) 33 06 63

CPS AT

CPS AT I
(siehe DOS-TEST 1/89)

80286-12 CPU 8/12 MHz, 1 MB RAM, bis 4 MB on board. 0-Wait-State, Sockel für Co-Prozessor, EMS, Phoenix Bios, 64 K ROM mit Setup-Funktion, Uhr u. Kalender, 1 ser./2 par./Game-Port, Herc. komp. Grafikkarte, 20 MB Festplatte 38 ms Zugriffszeit, 1 Lw 5,25" 1,2 MB und 1 Lw 3,5" 720 KB/1,44 MB Cherry Tastatur mit sep. Cursorblock, 14" Flat Screen Monitor s/w oder amber, MS-DOS 4.01 dt. und GW-Basic **3360,-**

CPS AT II

80286-16 MHz CPU. 12/16 MHz. 1 MB RAM, bis 4 MB on board, 0-Wait-State, EMS, Sockel für Co-Prozessor 80287, Bios 64 K ROM mit Setup-Funktion, Uhr u. Kalender, 1 ser./ 2 par./ Game-Port, VGA-Grafikkarte, 20 MB Festplatte 38 ms, 1 Lw 5,25" 1,2 MB u. 1 Lw 3,5" 720 KB/1,44 MB Cherry Tastatur mit sep. Cursorblock, 14" VGA Color Monitor MS-DOS 4.01 dt. und GW-Basic **4360,-**

CPS 386 SX
(siehe DOS-TEST 1/90)

80386 SX CPU 16 MHz, 0-Wait-State, 2 MB RAM, bis 4 MB on board, Sockel für 80387, VGA-Grafikkarte. 40 MB Festplatte 28 ms 1 ser./1 par./ Gameport, 1 Lw 5,25" 1,2 MB und 1 Lw 3,5" 720 KB/1,44 MB Cherry Tastatur mit sep. Cursorblock, 14" VGA Color Monitor MS-DOS 4.01 dt. und GW-Basic **4955,-**

CPS AT III / Cache

80386-25 MHz CPU. 20/25 MHz. 0-Wait 2 MB RAM, erw. bis 16 MB on board, 32 KB Cache. Memory Expans. Card Sockel für Co-CPU 80387, EMS-fähig, 2 par./2 ser. Schnittst./ Gameport, 1 Lw 5,25" 1,2 MB und 1 Lw 3,5" 1,44 MB, VGA Grafik EIZO MD-B 10, SCSI Combi Controller 157 MB SCSI Festpl. 14 ms 14" EIZO 9070s Monitor MS-DOS 4.01 und GW-Basic UNIX V/386 OS **15.500,-**

Technische Änderungen vorbehalten. Gerätekonfigurationen nach Ihren Wünschen veränderbar.

CPS-Filialbetriebe:

CPS Computertechnik GmbH
Braunhirschstraße 29, 3100 Celle
Tel. 0 51 41 / 3 20 04, Fax 0 51 41 / 38 14 22

CPS Computertechnik GmbH
Großbeerenstraße 5 · D-1000 Berlin 42
Telefon 0 30 / 705 30 55

CPS AT

80286-12 CPU 8/12 MHz, 640 KB RAM bis 4 MB on board, 0-Wait, 1 ser./ 2 par./ Game-Port, Herc. komp. Grafikk., 1 Laufw. TEAC 1,2 MB 5,25", Cherry MF II Tastatur 1880,-

Gerätekonfiguration nach Ihren Wünschen veränderbar. Speichererweiterungen bitte Tagespreis anfragen.

COMMODORE

AMIGA 500	995,-
AMIGA 500 + 1084	**1525,-**
AMIGA 2000	1985,-
AMIGA 2000 + 20 MB Harddisk Autoboot inkl. Contr. 2090 A	2990,-
AMIGA 2000 + Monitor 1084	2480,-
68020 Prozessorkarte	Preis a. Anfr.
68030 Prozessorkarte	Preis a. Anfr.

ERWEITERUNGEN

20 MB Harddisk A 2000 inkl. 2090 A Contr.	1220,-
2 MB Erw. int. (A 2000) orig. CBM	1180,-
PC/XT Karte inkl. 5,25" LW + MS-DOS + GW-Basic	899,-
PC/AT Karte inkl. 5,25" Lw + MS-DOS + GW-Basic	2400,-
512 KB Erw. A 500	330,-

AMIGA-ZUBEHÖR

LW ext. 3,5" ohne Display	290,-
LW ext. 5,25" ohne Display	419,-
LW intern 3,5" inkl. Einbausatz	220,-
Commodore 1084	630,-
Philips RGB Color CM 8833	630,-
EGA Standard	815,-

DRUCKER

OKI Microline 390 Centr.	Sonderpreis auf Anfr.
OKI Microline 393 Centr.	Sonderpreis auf Anfr.
OKI Microline 320	Sonderpreis auf Anfr.
OKI Microline 380	1138,-
OKI OL 400 + 800	Sonderpreis auf Anfr.
NEC P 6 Plus Centr.	1798,-
NEC P 7 Plus Centr.	1985,-
NEC Silentwriter LC 890	6995,-
Star LC 10 Centr.	448,-
Epson LX 400	448,-

Sheet Feeder

OKI 320/390	auf Anfrage
NEC P 6 +	541,-
NEC P 7 +	705,-
Centronics Druckerkabel	18,80

Weitere Drucker auf Anfrage

Wir liefern **nur** mit dt. Handbuch, Seriennummer und Herstellergarantie!!! Drucker-Grauimporte mit engl. Handbuch, ohne Seriennummer, ohne Herstellergarantie sind bei uns ausgeschlossen.



weil Preis und Leistung stimmen!

MONITORE

EGA Standard	815,--
EIZO 9060s	1855,-
NEC MULTISYNC II A	1550,-
TTL 14" Flat Screen mit Fuß sw/amber	235,-
TTL 12" Flat Screen mit Fuß sw/amber	180,-
Mitsubishi 1481	1398,-

weitere Monitore auf Anfrage

DISKETTEN

NN 2DD 3,5" 10 Stück	15,00
NN 2DD 5,25" 10 Stück	6,90
Magix MF 2DD 3,5"	24,50
Select MF 2DD 3,5"	33,50
Select MD 2DD 5,25"	19,00
Fuji MD 2DD 5,25"	27,00

FARBBÄNDER

Kores

STAR NL/NG/ND/NR-10, Stück	8,60
EPSON LX-800/LQ-500, Stück	7,50
PANASONIC KX-P, Stück	9,25
OKI ML 320	7,20
OKI ML 390	10,40
NEC 2200	10,25
NEC P6+/P7+	10,90
Star LC 10	7,00

SOFTWARE

ALDUS Pagemaker 3.0	2150,-
GEM Artline 1.0	1390,-
GEM Desktop 1.1	1200,-
MS-DOS Lernprogramm 2.0	168,-
MS-Windows 286 2.1	480,-
MS-Windows 386 2.1	750,-

weitere Software auf Anfrage

Sämtliche Angebote freibleibend, Zwischenverkauf vorbehalten. Wir liefern an Nicht-Kaufleute nur per UPS-Nachnahme. Ins Ausland nur per Vorkasse.

Fordern Sie unseren Gesamtkatalog gegen 3,- DM in Briefmarken.

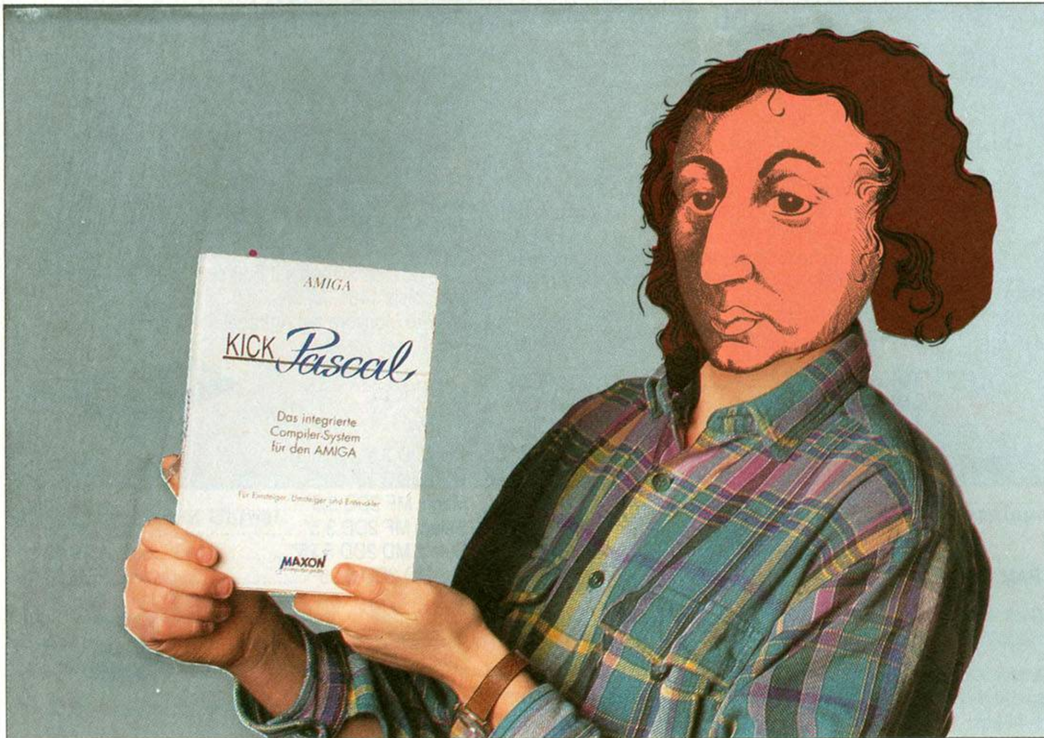
Es gelten unsere allgemeinen Geschäftsbedingungen.

Versand nur über Braunschweig!

- **24 Monate Garantie auf CPS-Rechner**
- **10 Tage Rückgaberecht auf Hardware**
- **Eigener Reparatur-Service**

Nutzen Sie unseren Bequem-Kauf-Kredit!

Die von uns angegebenen Taktfrequenzen (MHz) beziehen sich **nicht** auf den Speedtest nach Landmark, sondern sind **tatsächliche** Taktfrequenzen!



Michael Anton

Endlich Pascal für den Amiga!

Der Hoffnungsträger KickPascal im Test...

Fragt man Amiga-Programmierer, in welcher Sprache sie programmieren, geben die meisten "C" und "Assembler" an. Neuerdings kommt auch GFA-BASIC zur Sprache, wohl deswegen, weil es inzwischen einen Compiler dazu gibt. Modula2 findet auch seine Freunde, und AmigaBASIC... nun ja. Eine Sprache jedoch mußte bislang ein Schattendasein auf unserem 68000er fristen, und das, obwohl sie bei den PCs eine der Programmiersprachen schlechthin ist: Pascal. Das soll nun anders werden, und so bekamen wir kurz vor Redaktionsschluß noch ein Paket auf den Tisch. Der Inhalt? 'KickPascal'!

Was macht man eigentlich als eingefleischter Pascal-Programmierer auf dem Amiga? – Man ärgert sich, daß es die Firma Borland (bekannt durch TurboPascal auf dem PC) bislang noch nicht für nötig befunden hat, auch für den Amiga eine Version ihres sehr guten Pascal-Compilers zu entwickeln, während es beim PC schon eine Version 5.5 gibt. Als nächstes sucht man nach Alternativen, die man beispielsweise als PCQ-Pascal in der Public Domain findet. Der Wahre Jakob ist dieses noch nicht, an die rustikale Bedienung könnte man sich ja notfalls noch gewöhnen – mit Geschwindigkeit und Effizienz kann dieses Pascal jedoch nicht überzeugen. Auch der Funktionsumfang ist für den Praktiker nicht immer brauchbar, man denke hier an den etwas merkwürdigen Umgang mit REAL-Zahlen. Also her mit Sidecar oder Janus-Karte, um wenigstens unter MS-DOS ein brauchbares Pascal genießen zu können? Oder gleich C oder Modula lernen? Das muß nicht mehr sein, denn mit 'KickPascal' von Maxon ist der erste Schritt zu einer gelungenen Implementation von Pascal auf dem Amiga getan.

Pascal in Reinkultur...

Der erste Eindruck ist immer der wichtigste. Und der von KickPascal ist schon mal erfreulich. Nach dem Start aus der Workbench oder dem CLI meldet sich ein Editor zu Wort, der Dreh- und Angelpunkt des integrierten Systems von KickPascal ist. In Sachen Komfort kann er zwar nicht mit CygnusEd oder ähnlichen Editoren mithalten, für kleine bis mittlere Quelltexte reicht er jedoch allemal. Über die übliche Menüleiste oder optionale Tasten können nicht nur die Editorfunktionen bedient werden, sondern auch Parameter für den Compiler eingestellt und der Compiler selbst gestartet werden. Da sich der Quelltext und die Programmteile bereits im Speicher befinden, geht die Compilation sehr schnell vonstatten, da normalerweise keine Diskettenzugriffe mehr nötig sind. Lediglich beim Zugriff auf Include-Files muß noch 'genudelt' werden, aber auch diese Zeiten lassen sich durch eine wirklich geniale Art der File-Verwaltung minimieren: Sie können bei ihrer ersten Verwendung automatisch auf einen schnelleren Datenträger, beispielsweise RAM, kopiert werden und stehen bei den nächsten Durchläufen auch dort zur Verfügung, was vor allem in den Entwicklungsphasen spürbaren Zeitgewinn bringt.

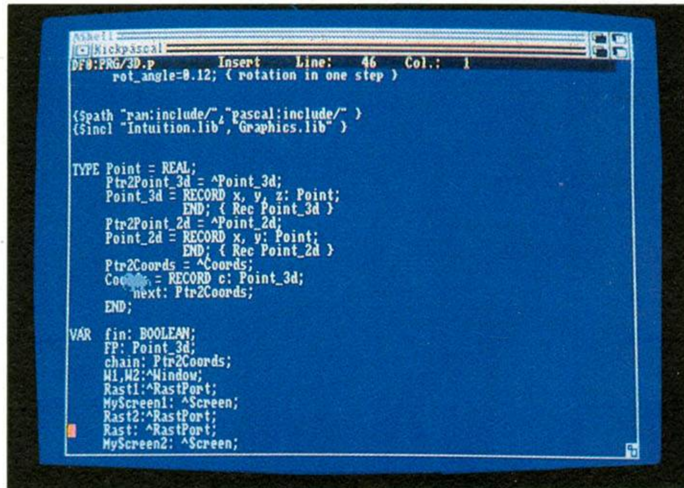
Unser Tip deswegen gleich hier: Wer genug RAM zur Verfügung hat, sollte sich eine Startdiskette von KickPascal erstellen, die alles ins RAM kopiert und den Editor von dort startet. Das Arbeiten mit dem Compiler wird dann fast zum Kinderspiel.

Das compilierte Programm kann nach dem Durchlauf entweder innerhalb des Editors getestet oder als ausführbare Datei abgespeichert werden. Bei Fehlern während der Compilation wird die dubiose Stelle im Quelltext angesprungen, bei Laufzeitfehlern kann diese gesucht werden. Der erzeugte Code ist schnell und relativ kompakt, der Overhead für das Runtime-Modul beträgt knapp 4 kByte. Abstürze während der Entwicklungsphase führen nicht automatisch zum Verlust des Quelltextes, hier schaffen eine Autosave-Option und ein Tool zum nachträglichen Retten des Textes genügend Sicherheitsreserven. Der Sprachumfang entspricht dem vollständigen Standard nach Jensen / Wirth und wird durch die Zugriffsmöglichkeit auf die System-Libraries des Amiga erweitert, für die Include-Files mit allen nötigen Definitionen mitgeliefert werden.

Das Handbuch – mal gestaunt und mal geflucht

KickPascal erlaubt neben der standardmäßigen Pascal-Programmierung auch die System-Programmierung. Das Erstellen von Screens und Windows, Gadgets, Menüs, Einbindung der Libraries und Intuition runden das gute Gesamtbild des KickPascal ab. Ein kleiner Schwachpunkt ist hier allerdings das Handbuch; es ist zwar alles recht

gut und locker (stellenweise recht amüsant) geschrieben, leider aber an einigen Stellen (vornehmlich bei Amiga-spezifischen Programmierungen) wieder nur für Profis verständlich. Nicht jeder weiß gleich mit Pointern und Strukturen was anzufangen. Hier wäre es wünschenswert, wenn entweder ein Einsteiger-Kurs beigelegt oder weitergehende Literatur erscheinen würde. Vor allem deshalb, weil KickPascal ja nicht nur für Umsteiger, sondern auch für Einsteiger interessant sein soll.



Der KickPascal-Editor mit einem Programm. Das Compilieren geht schnell, Fehler werden fast immer aufgefangen, solange man sich an die Programmierregeln hält. KickPascal hat eine gute Chance, seinen Platz bei den Programmiersprachen zu finden

Etwas schattig...

Die Konzeption als integriertes System bringt jedoch auch einige Nachteile mit sich. Wer nur in Pascal programmiert, legt wohl keinen großen Wert auf eine Linkfähigkeit, aber es soll ja auch Multilinguisten geben. Der Kontakt zu anderen Hochsprachen oder zu Assembler ist jedenfalls nicht möglich, allenfalls auf Umwegen über Libraries. Hier wäre zumindest die Integration von Assembler, entweder per Inline oder Mnemonics, wünschenswert. Auch wird die Unterstützung von größeren Programmen oder die modulare Programmierung etwas vernachlässigt. Die Verkettung von Programmteilen durch Chaining oder Overlays ist nicht möglich, das Aufteilen des Quelltexts in kleine Einheiten wird dadurch erschwert, daß bei Fehlern in Include-Dateien die betreffenden Stellen nicht in den Editor geladen werden. Die Dokumentation ist etwas dürftig, zur Ergänzung sind auf alle Fälle zusätzliche Bücher notwendig – sowohl in Sachen Pascal als auch zur Systemprogrammierung.

Fazit

Trotz dieser Schwächen, die sich in einer der nächsten Versionen wohl schnell ausbessern lassen, ist KickPascal ein sehr empfehlenswerter Compiler für alle, die 'null Bock' auf BASIC oder C haben und sich mit der finanz- und tippintensiven Arbeit des Modula noch nicht anfreunden konnten. Der Pascal-Notstand auf dem Amiga ist mit KickPascal nun vorerst behoben, es ist zu hoffen, daß die Sprache auch auf dem Amiga weite Verbreitung (natürlich nicht als Raubkopie!) finden wird...

(jb)

```
{Kleines Testprogramm der AMIGA DOS by jb}
{erhebt keinen Anspruch auf den ersten Preis}
{fuer sauberes Programmieren...}

Program Show_it;
   {$path "include/"}
   {$incl "graphics.lib","intuition.lib"}

Const xEcke = 0;
      yEcke = 0;

Var   MyNewScreen       : Ptr;
      MyNewWindow       : ^Window;
      x1,y1,x2,y2       : Integer;
      Satz              : IntuiText;
      Anzahl            : Integer;

Begin {AMIGA DOS - Test}
   OpenLib(IntBase,'intuition.library',0);
   MyNewScreen := Open_Screen(  0,  0, {Anfang des Screens }
                              640,256, {Ende des Screens    }
                                4,     {Tiefe --> Bitplanes}
                               14,  5, {Farbe Schrift,Hint.}
                                $8020, {Modus               }
                'Ein neuer Screen...'); {Screen_Text         }

   MyNewWindow := Open_Window(  0, 50, {Anfang des Windows }
                              640,100, {Ende des Windows    }
                                $0908, {Farbe --> Hi=Schriftfarbe Rahmen}
                                       {Farbe --> Lo=Rahmenfarbe  Rahmen}
                                $0400, {Intuition Direct Communication Message Port}
                                $1001, {Window - Flags}
              '...mit neuem Window...', {Window - Name }
                              MyNewScreen,          {Screen zum Window}
                              100,     {Koordinaten, die beim 'ziehen'   }
                              100,     {mit der Maus nicht unterschritten}
                              80,      {werden sollten, das Window kann  }
                              30);     {also nicht kleiner werden!!!     }

   Satz := IntuiText( 4, 0,    {FrontPen, BackPen}
                         1,    {Drawmode         }
                      0,10,    {LeftEdge,TopEdge }
                       Nil,    {TextFont normal  }
'...und einem Text, der ueber den Bildschirm wandert...', {Message}
                       Nil);   {Zeiger auf weitere Texte}
   Anzahl := 600;

   While Anzahl > -420 Do
      Begin { While }
         IF (Anzahl < 599) OR (Anzahl > 10) THEN
            PrintIText(MyNewWindow^.Rport, {Zeiger auf Rastport}
                  ^Satz,                   {Zeiger auf auszugebenden Text}
                  Anzahl,                  {Koordinaten auf Window-Position}
                  40);
         Anzahl := Anzahl - 2;
      End; { For }
   Close_Window(MyNewWindow);
   Close_Screen(MyNewScreen);
End.
```

Listing

Kennen Sie KickPascal?

Wenn ja, dann schreiben Sie uns doch mal Ihre Meinung über eine Pascal-Ecke in der AMIGA DOS. Unserer Meinung nach verdient KickPascal es, sich unter die vorhandenen Programmiersprachen zu mischen, vor allem deswegen, weil es auch hier möglich ist, System-Programmierung einzubinden. Wenn Sie also auch Pascal-Programme in der AMIGA DOS finden wollen, schreiben Sie uns.

AMIGA DOS Blitzlicht

Name: KickPascal

Hersteller, Vertrieb: Maxon

Hardware: alle Amiga ab Kickstart 1.2, mind. 1 Floppylaufwerk und 512 kByte RAM (mehr ist besser!)

Preis: 189 DM

Positiv:
- integriertes System
- sehr schnell und kompakt
- vollständiger Standard
- intelligente Include-Verwaltung

Negativ:
- nicht linkfähig
- keine Assemblereinbindung
- Dokumentation etwas dürftig

AMIGA DOS

Alle Programme sofort nutzen

Da ist er nun endlich – der Trick oder das Programm, auf das Sie schon so lange gewartet haben! Zu allem Unglück ist das Listing aber in Modula-2 oder C, jedenfalls in einer Sprache, zu der Sie keinen Compiler haben, um ein lauffähiges Programm herzustellen.

Auch in diesem Fall hilft Ihnen die Databox von Amiga DOS aus der Patsche: Neben den Quelltexten im ASCII-Format finden Sie jeweils auch das fertige, lauffähige Programm. Sie brauchen es nur von der Databox-Diskette aus zu starten.

Endlich keine Listings mehr abtippen!

Nicht bei allen Programmen ist es mit drei Zeilen getan – gute Routinen und praktische Funktionen brauchen ihren Platz. Und bisweilen lassen sich auch lange Datenblöcke nicht vermeiden, ganz zu schweigen von Hexdumps und Assemblerlistings.

Schonen Sie Ihre Augen und schlagen Sie sich nicht die Nacht mit Abtippen um die Ohren. – Auf der Databox zum Amiga DOS-Heft finden Sie alle Listings als ASCII-File: passend für jeden Texteditor, den Amiga-BASIC-Interpreter, Makro-Assembler oder einen Compiler für C und Modula-2.

DATA

Alle Listings und Programme auf Diskette –



Sofort anfangen – ohne Zeitverlust!

Für was Sie sich auch interessieren, Interna des Amiga, Anwendungen, oder die Beispielprogramme der Werkstätten im Amiga DOS-Heft: Bevor Sie richtig an die Arbeit gehen können müssen Sie erst die benötigten Programme eingeben, korrigieren und zum Laufen bringen.

Sparen Sie sich diesen ermüdenden, unnötigen Aufwand an Zeit und Konzentration. Mit der Databox für Ihr Amiga DOS-Heft können Sie sofort den Dingen auf den Grund gehen, die Sie wirklich interesssieren.

Keinen Ärger mehr mit Tippfehlern

Wer kennt das nicht, wenn das Programm nach dem Eintippen nicht läuft oder der Rechner gar abstürzt. Besonders gemein sind auch Fehler, die erst nach Wochen bei einer bisher nicht gebrauchten Funktion zu Tage treten, oder wenn der Druckfehlerteufel am Werke war.

Zermartern Sie sich nicht den Kopf, bis Sie die falsche Zahl im Datafeld gefunden haben. – Alle Dateien auf der Databox zur Amiga DOS sind vom Autor und der Redaktion auf Fehlerfreiheit geprüft und im dazugehörigen System "probegelaufen".

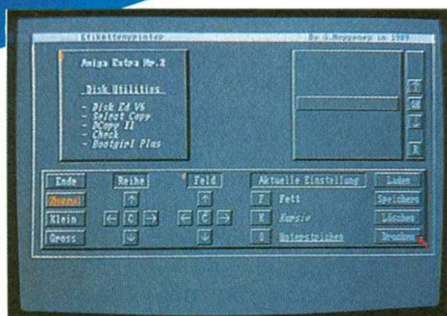
2 Disketten voller Programme

BOX

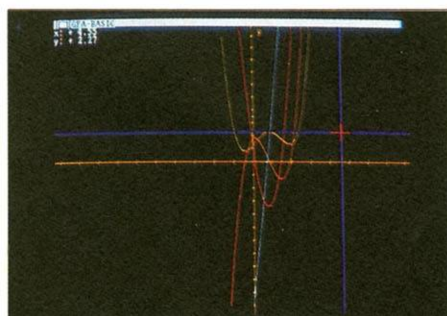
Computer einschalten – Diskette einlegen – los geht's



Schach – Das Listing des Monats. Dazu gibt es gratis Spielfiguren in Super-Grafik



Diskettenlabels selbst gestalten – mit unserem Programm Etikettenprint kein Problem



Mathematische Kurven und deren Ableitungen grafisch darstellen – FunkPlot macht's möglich

Inhalt der Databox AMIGA DOS 3/90

Basic	C	Modula-2	Assembler	Lauffähiges Programm	
X				X	BASIC-Kurs – alle Beispielprogramme lauffähig auf der Databox
X				X	Diskettenaufkleber im Eigenbau
X				X	Gregor – Berechnung von Wochentagen in GFA-BASIC
X				X	Funktionsplotter in GFA-BASIC
	X			X	Das Sprite-Editor-Projekt – Der zweite Teil zur professionellen Erstellung von Sprites, BOBs und Images
		X		X	Die Modula-Werkstatt: nützliche Tips&Tricks in M2Amiga
			X	X	Ein Schachprogramm, bei dem man seine Figuren mit jedem IFF-fähigen Grafikprogramm selbst gestalten kann
			X	X	Beispielprogramme zur Blitter- und Copperprogrammierung
			X	X	Aufsteigerwissen Assembler: Paintbox, der dritte Teil unseres Grafikprogramms

24,– DM

Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

Inland:		Ausland:	
Einzelpreis	24,– DM	Einzelpreis	24,– DM
zzgl. Versandkosten	4,– DM	zzgl. Versandkosten	6,– DM
Endpreis	**28,– DM**	**Endpreis**	**30,– DM**

Zahlungsweise:
Am einfachsten per Vorkasse (Verrechnungsscheck) oder als Nachnahme zuzüglich der Nachnahmegebühr. (Bei Lieferungen in das Ausland ist Nachnahme nicht möglich.)

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege





Bild 1: TURBOprint II bietet eine Menge Einstellmöglichkeiten für den Drucker, die weit über die Preferences-Einstellungen reichen

Druckertreiber

Falls der Drucker nicht will – TURBOprint II

Drucker und Amiga – die beiden legen bei so manchem Zusammentreffen Starallüren an den Tag, daß dem armen Besitzer Hören und Sehen vergeht. Was man dagegen tun kann? Entweder still leiden – oder, wenn es nach dem Wunsch der Firma Irsee geht, 'TURBOprint II' kaufen. Wir haben uns für Sie mit dem Programm einmal näher befaßt.

Es hapert bei der Ansteuerung eines Druckers vom Amiga aus, vor allem dann, wenn der eigene Drucker nicht eines der 'Preferences'-Wunschgeräte ist. TURBOprint II will hier nun den Part der hauseigenen Druckerinstallation übernehmen, ein Vorhaben, das zwar auf den ersten Blick lobenswert ist, gleichzeitig aber diejenigen, die für eine Hardcopy den ganzen Nachmittag benötigen, zweifeln läßt. Um eines gleich vorwegzunehmen: Den Zweifel können Sie vergessen.

Drucker sind zum Drucken da, und nicht zum Ärgern – Maxime von TURBOprint

TURBOprint II meldet sich nach dem Start mit einem Workbench-Screen und dem Disk-Icon. Auf der Diskette befindet sich neben dem TURBOprint-II-Programm noch ein 'NoTurbo-', ein 'Install-' und ein 'Readme'-File.

Das Install-File plaziert TURBOprint II auf die Festplatte, so vorhanden. Im ReadMe-File sind Ergänzungen zum Handbuch aufgezeigt, darunter Tabellen, in denen für verschiedene Drucker Ersatz-Treiberprogramme aufgezählt sind. Wer nach dem Starten des Programms noch immer Schwierigkeiten hat, sollte auf jeden Fall erst mal in dieses ReadMe-File hineinschauen, hier kann man immer noch den Rettungsanker finden, falls fast alles versagt.

'TurboPrefs' schließlich ist das eigentliche Programm. Hier muß nun der Druckerbesitzer seinen Drucker wiedererkennen. Das heißt, anhand der internen Werte muß ermittelt werden, ob der Drucker auf die ausgesandten Steuerzeichen so reagiert, daß es zu keinen Fehldrucken kommt.

TURBOprint II hat intern schon eine Menge Treiber integriert, unter anderem für die meistbenutzten EPSON-, NEC-, CANON-, HP-, OKIDATA-, PANASONIC-, SEIKOSHA-, SIEMENS- oder STAR-Drucker. Man sieht schon, die Erzeuger des Programms haben sich eine Menge Gedanken gemacht, wie man geplagten Drucker- und Amiga-Usern helfen kann.

Den eigenen Drucker findet man schnell im Directory mit den Firmennamen. Hier kommt jetzt auch das ReadMe-File zur Geltung. Findet man seinen Drucker nicht unter den angegebenen, ist es immer noch möglich, einen Ersatztreiber anzuwählen. Dieser Ersatztreiber läßt sich der Tabelle des ReadMe-Files entnehmen. Sollten alle Stricke reißen, arbeitet TURBOprint II auch mit den 'Ur'-Treibern der Preferences zusammen,. dieser wird durch Eingabe des Pfadnamens im Druckerauswahl- Menü eingestellt.



Bild 2: Im 'Prefs'-Menü läßt sich der Drucker auf einen Grafikausdruck vorbereiten. Hier ist Experimentieren gefragt, klappt es nicht gleich beim ersten Mal, kommt Versuch Nummer zwei...

Hat man den Treiber für seinen Drucker gefunden, so ist es sinnvoll, sich jetzt Gedanken über den möglichen Grafikausdruck zu machen. TURBOprint II kann dabei Textdruck, Text und Grafik auf einer Seite oder sogar Farbdruck (mit dem entsprechenden Drucker, versteht sich) in hoher Auflösung übernehmen. Aber, wie schon gesagt, erst probieren und testen.

Da der Amiga ein tolles Farbwunder ist, die meisten Drucker jedoch nicht, ist es nötig, den Farben ein bestimmtes Muster zuzuordnen, damit auf einem Ausdruck auch wirklich etwas zu sehen ist. Im Pattern-Menü läßt sich dieses auswählen, der Grafikausdruck bekommt dann diese Musterung zugeordnet, und die Farben werden hinterher von diesem Muster übernommen.

Der 'Aspect Ratio'-Regler dient zum Einstellen des Breiten-/Höhenverhältnisses, welches bei einigen Druckern nicht von vornherein stimmt, sondern erst eingestellt werden muß. Zu merken ist das vor allem dann, wenn die Tortengrafik nicht wie Torte, sondern eher nach Eiergrafik aussieht, durch Regeln des Höhen-/Breitenverhältnisses läßt sich dieses Problem umgehen.

Wundern Sie sich nicht, wenn Sie im Screen einen Helligkeits- und einen Kontrastregler für den Drucker finden. Das Ganze dient dem Zweck, zu dunkle Farben beim Ausdruck aufzuhellen, bei unserem Probeausdruck (Abb.1) war nämlich in der inversen Darstellung fast alles zu schwarz, so daß kaum noch Details zu finden waren. Nach dem Einregeln in einen helleren Bereich wurde der Ausdruck tatsächlich besser. Es kann allerdings auch der gegenteilige Effekt auftreten, daß nämlich zu helle Graubabstufungen, die durch das eingestellte Muster repräsentiert werden, ineinander überfließen. Auch hier ist das Probieren angesagt. Die anschließende Farbkorrektur bringt es an den Tag, welche Farben zu schwach oder zu stark ausgedruckt werden, auch hier hilft das Nachregeln.

Beim Aufruf von 'Call Prefs' sieht man noch mehr Einstellmöglichkeiten für den Drucker im 'Extended Prefs-Menü'. Hier kann man die Art des Ausdrucks wählen, wie

Color (ist natürlich nur bei Farbdruckern sinnvoll), Gray-Scale (dazu dient das angewählte Muster) oder Black & White (schwarzweiß, also nur mit einer Farbe). Mit 'Image' kann ein normaler oder invertierter (positiver oder negativer) Ausdruck gewählt werden, 'Aspect' dient zum Anwählen der horizontalen oder vertikalen Druckrichtung. Was jetzt noch an Einstellmöglichkeiten da ist, dient nur noch den Feinheiten beim Ausdruck.

Eines wäre noch zu erwähnen: Antialiasing. Hinter diesem Zungenbrecher verbirgt sich eine Glättefunktion. Da Hardcopies den Bildschirmen entsprechen, Bilder aber auf Pixel basieren, sehen manche Rundungen nicht rund, sondern abgehackt eckig aus. Besonders bei speziellen Zeichensätzen gibt dies kein schönes Gesamtbild. Durch 'Antialiasing ON' werden diese Ecken einer Glättefunktion unterworfen, die den Ausdruck wesentlich glatter erscheinen läßt.

Das Tolle an TURBOprint II ist, daß es eine Hardcopy-Funktion enthält, die resetfest und per Drei-Tasten-Druck aufrufbar ist. Mit dieser Funktion ist auch das Bild aus Leisure Suit Larry entstanden. Diese Hardcopy-Funktion blieb tatsächlich eine geraume Zeit im Speicher, obwohl mehrere Programmdisketten gebootet wurden.

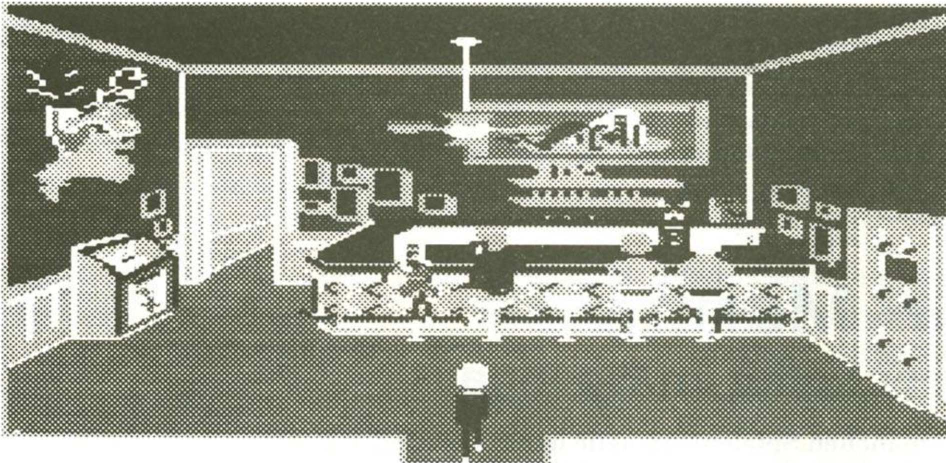
So sollte ein Druckertreiber sein: Den Drucker treibend und per Taste präsent

Das Menü selbst ist ebenfalls durch Tastendruck wieder aufrufbar, ohne von Diskette nachgeladen werden zu müssen. Überhaupt sind eine Menge Tastenfunktionen im Programm enthalten: Es lassen sich Bilder auf Diskette abspeichern, Bilder vom Workbench-Screen lassen sich einfach in Schwarzweiß drucken, ohne das Farbband zu schädigen (der Unterschied zum Normalausdruck ist gewaltig), Bildteile können per Taste ausgeschnitten und in dieser Form gedruckt oder abgespeichert werden. Man sieht, TURBOprint II ist nicht nur ein spezielles Treiberprogramm, es ist darüber hinaus ein prima Utility mit Hardcopy-Funktion.

TURBOprint II verfügt auch über die Möglichkeit, Laser-Drucker anzusteuern, sofern diese grafikfähig sind. Hierzu gibt es zum Beispiel die EPSON GQ-Treiber.

Um TURBOprint II richtig zu nutzen, ist es auf jeden Fall notwendig, sich in das Programm einzuarbeiten. Auch, wie schon gesagt, kommt man nicht darum herum, ein paar Ausdrucke in den Papierkorb zu werfen und mit anderen Einstellungen noch mal zu beginnen. Jedoch wird man den Wert des Programms schnell erkennen, wenn der erste Ausdruck per Hardcopy gemacht wurde, an den man sonst nie herangekommen wäre. Somit steht als Urteil fest: Ein sehr willkommenes Utility.

(jb)



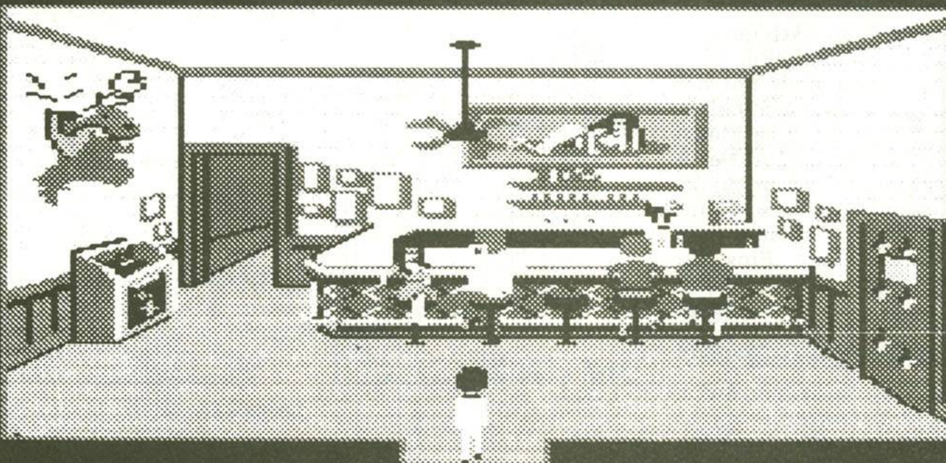


Abb. 1: Als Testobjekt mußte Larry Laffer (bekannter Sierra-Held) auf einem 24-Nadel-Drucker herhalten, oben als Inversdarstellung, unten in normaler Darstellung, beide Ausdrucke im Gray-Scale-Betrieb. Dank der residenten Hardcopy-Routine gab es hierbei keine Probleme.

AMIGA DOS Blitzlicht

Name: TURBOprint II
Hersteller: IRSEESOFT
Grüntenstr.6, 8951 Irsee
Tel.: 08341/74327
Funktion: Drucker-Utility
Preis: 98,- DM

Positiv:
- viele Druckertreiber
- resetfeste Hardcopy
- Bildausschnittfunktion für Speichern und Hardcopy
- Speichern von Bildern per Tastendruck
- 'FastMem Last'-Funktion für speicherprüfende Programme
- leichte Bedienung des Menüs per Maus
- Preferences-Treiber weiter verwendbar
- residentes Menü einstellbar
- auf Festplatte installierbar

Negativ:
- keine negativen Punkte feststellbar...

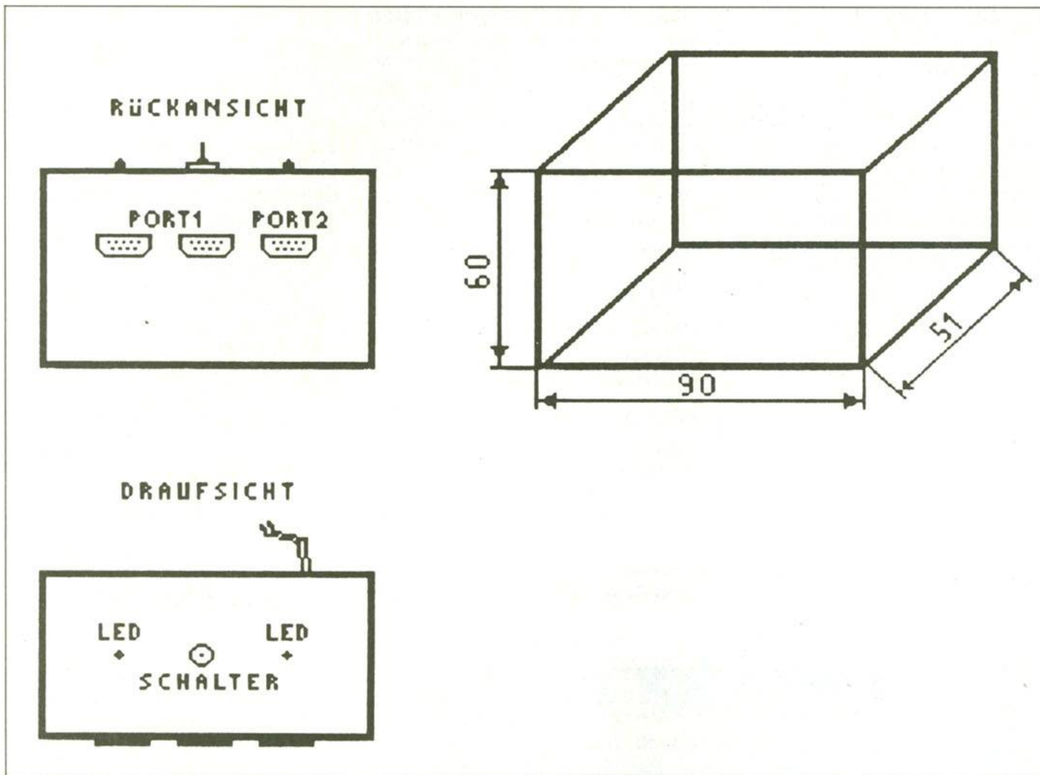


Abb. 1. Das "Drumherum" unseres Umschalters. An der Rückseite der Box sind zwei Ports parallel geschaltet.

Michael Trebing

Selbst ist der Mann...

Maus-/Joystickumschalter im Eigenbau

Wer kennt nicht das Problem bei manchen Spielen? Anstelle der Maus wird ein Joystick im Mausport benötigt. Da heißt es dann: Maus raus, Joystick rein. Nicht nur, daß dies recht umständlich ist, nein, auch die Ports leiern mit der Zeit aus, so daß eine feste Verbindung zwischen Maus bzw. Joystick und Computer nicht mehr einwandfrei gegeben ist. Unser Maus-/Joystickumschalter Marke Eigenbau soll nun Abhilfe schaffen.

Aber auch die Computerbesitzer, die nicht gerade viel Platz auf ihrer Arbeitsfläche haben, finden einen Vorteil in dem Umschalter. Durch ein extra langes Kabel kann eine Vergrößerung des Arbeitsbereiches erfolgen.

Das Gehäuse kann dann an einem beliebigen Platz postiert werden, von dem die Anschlüsse für Maus und Joysticks ausgehen.

Bei unserem Umschalter wird zusätzlich noch ein weiterer Port durchgeschleift, an dem beispielsweise ein Lightpen oder ein zweiter Joystick seine Arbeit verrichten könnte.

Wer auf Luxus steht, der kann zusätzlich zwei LEDs einbauen, die anzeigen, welcher Port momentan aktiviert ist. Der softwarekompatible Umschalter funktioniert mit der gängigsten Software auf allen Computern, die einen Maus- und einen Joystickport besitzen. Probleme könnten jedoch eventuell bei der Hardware auftreten: Wenn man nämlich spezielle Pins für Dongles belegen muß (z.B. Superbase), oder ein Gerät durch den Joystickport wichtige Befehle abgibt (z.B. BTX-Manager).

Die beiden Output-Ports 1a und 1b liegen spiegelverkehrt zueinander. Aus diesem Grund sollte man auf den korrekten Anschluß achten.

Die zwei Stecker werden in den Maus- und Joystickport gesteckt, während beispielsweise die Maus und zwei Joysticks permanent an der Umschaltbox betrieben werden können. Die Eingangssignale kommen von Maus- und Joystickport des Rechners. Der Schalter ermöglicht nun das Umschalten zwischen den Ports direkt an der Box.

Mit dieser einfachen und doch wirkungsvollen Konstruktion können Sie sich bequem in Ihren Sessel zurücklehnen und Ihr Lieblingsspiel genießen. Körperliche Verspannungen und der damit verbundene Gang zum Massagesalon dürften somit der Vergangenheit angehören.

(br)

Bei unserer kleinen Bastelei werden folgende Utensilien benötigt:

- ein Gehäuse (Abmessungen siehe Abbildung 1)
- ein zweipoliger Umschalter
- drei Joystickports
- zwei neunadrige Joystickkabel mit Stecker (Länge nach Bedarf)
- ein bis zwei Meter Stromlitze (je nach Bedarf)
- zwei LEDs (5 V)
- einen Widerstand (10 bis 100 Ohm)

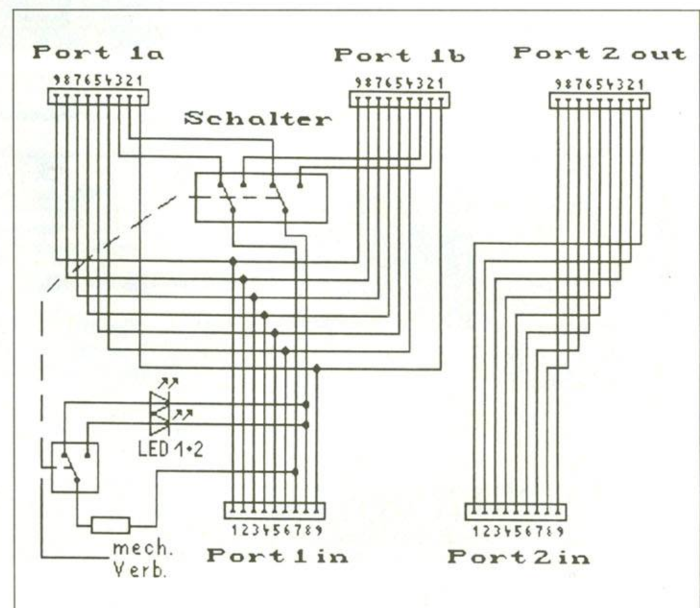


Abb. 2. Das Innenleben der "Kiste" – Ein vereinfachtes Schaltbild.

Verleger am heimischen Herd

Das DTP 'The Publisher' im Test

Der Amiga kann mehr, als unendlich viele Aliens auf den Bildschirm zu zaubern, die man mühevoll mit dem Joystick jagen muß. Dies haben inzwischen sehr viele Softwarehersteller erkannt, darunter auch Gold-Disk. 'The Publisher' nennt sich ein Programmpaket mit Textverarbeitung und Layoutprogramm, das den Amiga zu einer zeitungsmachenden Heimstation werden läßt. Ob der Publisher da nicht zuviel verspricht? Wir haben ihn getestet.

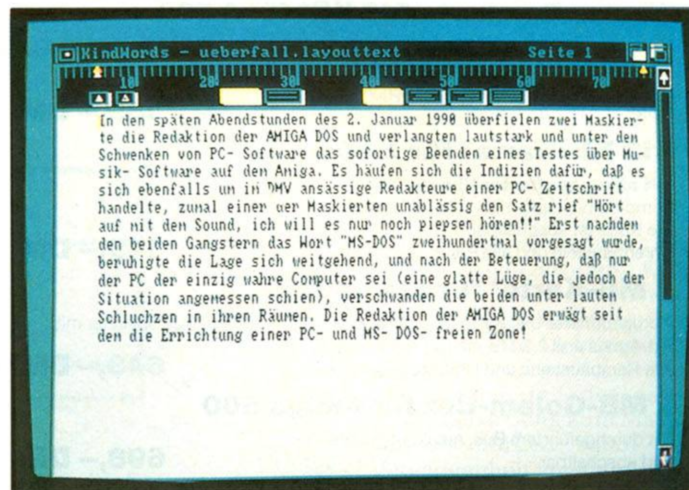


Bild 1: 'KindWords' heißt die dem 'Publisher'-Paket beiliegende Textverarbeitung. Texte für die Layoutseite können hier schnell geschrieben werden.

Das Programmpaket 'The Publisher' besteht aus sieben Disketten, zu denen außer der 'Kindwords'-Textverarbeitung mit SuperFonts und Dictionary-Disk noch das Layoutprogramm PageSetter mit FontPack-, ArtPack- und LaserScript-Diskette gehört, eine Komplettausrüstung also. 'KindWords' entpuppt sich als solide Textverarbeitung, mit allem, was zu einem solchen Programm gehört. Blockoperationen gehören ebenso dazu, wie Trennhilfe und Formatänderungen. Texte, die mit 'KindWords' bearbeitet wurden, lassen sich sofort vom Publisher nutzen.

'KindWords' ist der erste Schritt zum gedruckten Papier

Die Textverarbeitung kann ihre englischsprachige Herkunft nicht verleugnen, auch wenn Menüs und Hilfen in Deutsch sind. Ganz deutlich wird dies beim Aufruf des mitgelieferten Spell-Checkers, also der Rechtschreibprüfung. Vor allem Worte, die in Deutschland zur Umgangssprache gehören, oder fremdsprachliche Wörter werden vom 'Rechtschreibprüfer' in einer fast komischen Art interpretiert, die Ersatzwörter sehen dementsprechend aus (würden Sie vermuten, daß man Amiga durch Magda ersetzen kann?). Trotzdem ist 'KindWords' eine gute, wenn auch nicht allzu schnelle Textverarbeitung, die enthaltenen Funktionen sind bald erlernt.

Mit 'KindWords' erstellte Texte lassen sich auf zwei Arten speichern, einmal als reiner ASCII-Text, sonst im 'KindWords'-Layoutformat. Für den 'Pagesetter' kann das Layoutformat gewählt werden, da dieser direkt mit KindWords zusammenarbeitet. Was uns negativ an 'KindWords' auffiel, ist die teils schlechte Umsetzung ins Deutsche; Fonts, die mit 'griechlisch', betitelt werden, haben in einem Programmpaket dieser Art eigentlich wenig zu suchen.

Der Text ohne Bild ist wie ein Fisch ohne Wasser – trocken

'KindWords' ist das, was man allgemein als Standard-Textverarbeitung umschreibt. Um nun aus den Texten ein ansprechendes Layout-Dokument zu machen, wird der 'PageSetter' aufgerufen.

'PageSetter' sieht auf den ersten Blick etwas ungewöhnlich aus. Die zu erstellende Seite wird komplett auf dem Bildschirm dargestellt, natürlich noch ohne Inhalt. Ein Lineal sorgt senkrecht und waagerecht für korrekte Positionierung, läßt sich jedoch auch ausblenden, falls dies erwünscht ist.

Jede Text- oder Bildstelle muß durch eine Box repräsentiert werden, indem das Gewünschte seinen Platz findet. Ist diese Box einmal belegt, so läßt sich dort nichts mehr eintragen, nur durch geschicktes Plazieren einer neuen Box oder gesamtes Löschen der alten ist also ein Plazierungsfehler wieder zu beheben.

Die Seitenmaße können übrigens frei gewählt werden, vorhanden sind die amerikanischen Maße (in Inch) 8,5x11 und 8,5x14 sowie die DIN-Maße A4 und A6. Das Lineal kann auf die Skala 1/8", 1/6" oder PICA, einer Maßeinheit aus dem Zeitungswesen, eingestellt werden. Zusätzlich kann noch zur Positionierungsunterstützung ein Gitter in die Seite integriert werden, dessen Maße ebenfalls wählbar sind.

Boxen können auch automatisch erstellt werden, indem man den Text direkt plaziert. Diese Methode ist jedoch nicht unbedingt sinnvoll, vor allem dann, wenn man ein bißchen Wert auf das auszugebende Dokument legt. Hier ist es in jedem Fall sinnvoller, sich die Mühe zu machen, jede Box per Hand zu erstellen, auch wenn man ab und zu von vorne beginnen muß.

Bei der Positionierung der Boxen muß man aufpassen, daß man den eingestellten Seitenbereich nicht verläßt, sonst kann es passieren, daß der rechte Rand nicht vollständig ausgedruckt wird. Um Text und Grafiken einzulesen, besitzt 'Pagesetter' einen eigenen Text- und einen eigenen Grafikeditor. Der Texteditor ist fast selbst wie eine Textverarbeitung, Texte können dort auch eingegeben werden, jedoch fehlen ihm sämtliche Steueranweisungen, wie Schriftarten, Absatz- und Formatkennungen. Texte aus anderen Textverarbeitungen als den von 'PageSetter' angegebenen sollten also im ASCII-Format vorliegen, denn sonst werden die Steuerzeichen als normale Zeichen ausgegeben, und dies ist dem Text nicht gerade dienlich.

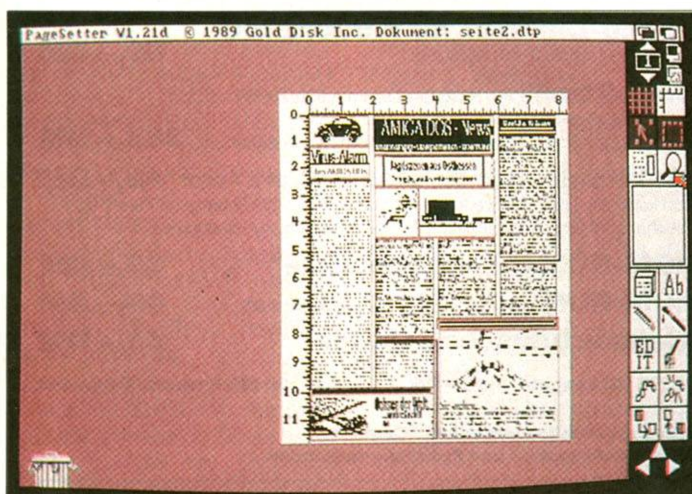


Bild 2: 'PageSetter' ist das eigentliche Desktop-Publishing-Programm. Hier wird die Seite so 'zusammengeschnibbelt', wie man sie später auf Papier sehen will. Davor jedoch liegt eine Menge Arbeit.

Die Texte lassen sich jedoch immer noch im Schriftbild verändern, diverse eigene Steuerzeichen müssen im Text eingegeben werden, um zum Beispiel eine Fett- oder Schattenschrift zu erzielen oder Trennungen im Text einzubauen, damit zwischen den Wörtern nicht zuviel Zwischenraum entsteht.
Ist der Text soweit fertiggestellt, so empfiehlt es sich, ihn komplett zu markieren und "herauszuschneiden". Diese Funktion dient üblicherweise dazu, Blöcke kenntlich zu machen, aus dem Text herauszunehmen und an anderen Stellen wieder einzusetzen. Das Markieren des gesamten Textes ist eine notwendige Maßnahme, denn so hat man ihn immer 'griffbereit'. Wird der Text nämlich in die Box übernommen, enthält er einen Fehler, den man vorher nicht bemerkt hat und wird gelöscht, muß er noch einmal von Diskette geladen werden. Zwar gibt es beim Texteditor die Funktionen LOAD und SAVE, das Ganze gestaltet sich jedoch umständlich, wenn man jedes Mal den Text erst aus den 'Schubladen' einer extra erstellten Diskette hervorkramen muß.

Ist die Box einmal belegt, kann man nichts mehr ändern! Daraus folgt: Vorher überlegen

Nach dem Übernehmen des Textes in das Layout sind nur noch Korrekturen der aktuellen Box möglich, der Text bleibt (bis auf die Schrift) so, wie er ist. Die Box kann jetzt noch nachgestaltet werden, das heißt, es können Abgrenzungsrahmen an den Seiten, oben und unten oder um die ganze Box herum gesetzt werden. Außerdem kann der Font (Zeichensatz) geändert werden, um ein besseres Schriftbild zu erzielen. Ferner ist es möglich, der Box ein Hintergrundraster zur Unterlegung zu geben. Da die Schrift auch invers dargestellt werden kann, läßt sich also auch ein schwarzer Hintergrund wählen.

Leider zeigen sich gerade bei der Box die Schwächen des PageSetters, wählt man eine kleinere Schrift als die normale 'Topaz 9' und dazu ein Hintergrundraster, verschwimmt das Ganze auf einem Ausdruck, der auf einem Matrixdrucker gemacht wurde. Nur bei Laserdruckern ist hier mehr zu erwarten. Dazu stimmen die Höhen der einzelnen Fonts nicht hundertprozentig, so ist teilweise das 't' größer als die meisten Großbuchstaben, was dem Schriftbild ein teilweise exotisches Aussehen verleiht.

Grafiken können von der beiliegenden 'Artist's Choice'-Disk geladen oder unter Mal- und Zeichenprogrammen erstellt werden. Das Einsetzen in das Layout ist ähnlich wie beim Text: Die Grafik wird durch einen Rahmen 'herausgeschnitten' und dann in die dafür bestimmte Layoutbox gesetzt. Danach läßt sie sich noch in allen vier Richtungen verschieben, so daß man den optimalen Platz dafür bekommt. Ein Tip: Überschriften und Aufmacher kann man ebenfalls im Grafikfeld machen, da sich hier auch kurze Textpassagen in allen Fonts erstellen lassen. Aber auch hier gilt: Einmal in eine Box gesetzt, läßt sich in eine Box nichts mehr hineinschreiben oder -zeichnen. Also ist vor allem hier vorher die Überlegung nötig, wie es hinterher aussehen soll.

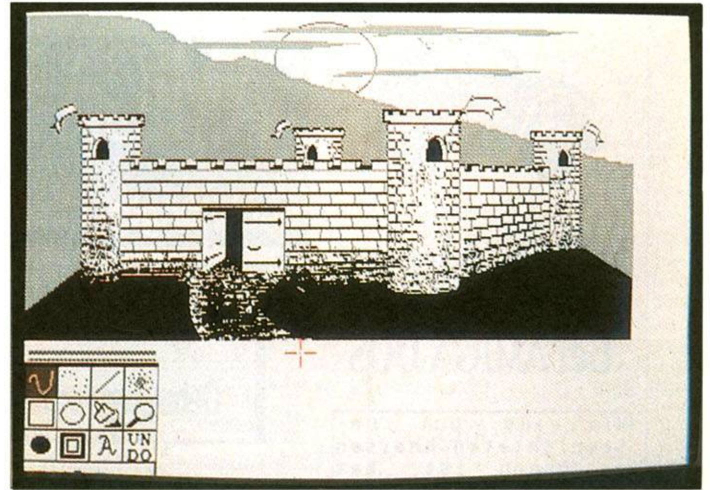


Bild 3: Der Grafikeditor von 'PageSetter'. Hier können Grafiken von der 'Art-Choice'-Diskette oder andere Grafiken für den DTP-Teil vorbereitet werden.

Jeder Amigabesitzer wird zum eigenen Verleger – macht es 'The Publisher' möglich?

Wir meinen, nein. Man muß sich beim 'Publisher' schon vor dem Kauf im klaren sein, wozu man das Programm benötigt. Seine Stärken zeigt das Programmpaket nur in Verbindung mit einem Laserdrucker oder, noch besser, falls Sie im Besitz von CompuGraphic-Satzmaschinen sind, und die sind für Normalanwender so sinnvoll wie ein Regenwurm als Köder für den weißen Hai. Die Möglichkeit, Layoutseiten in das PostScript-Format umzuwandeln oder die Laser-Script-Diskette für die Ausgabe auf grafikfähige Laserdrucker hinzuzuziehen, hebt den 'Publisher' in Richtung professionelle Anwendungen, durch die Probleme beim Ausdruck auf Matrixdruckern (Schriftbild und Zeichensätze) muß sich der 'Normal-Amigabesitzer' entscheiden, ob die Investition lohnt. 'The Publisher' als schlechtes Produkt zu bezeichnen, geht auch nicht. Und zwar deshalb, weil das Erstellen von Layoutseiten für private Zwecke Spaß macht. Seien es humorvoll mit Zeichnungen unterlegte Briefe, Einladungen zum nächsten Kegelabend oder Glückwünsche zum Geburtstag, die eine Schwarzweiß-Tortengrafik enthalten, eine derart erstellte Seite macht Eindruck. Jedoch muß man sich hier schon überlegen, ob der Einsatz des 'Publisher' rentabel ist. Auf jeden Fall kann er den Besitzer eines Amiga in seinem Verein aufwerten.

(jb)

AMIGA-DOS-News – von der Idee zum Ausdruck

Ein Desktop-Publishing-Programm zu testen, ohne es selbst probiert zu haben, ist nicht gerade sinnvoll. Was also bietet sich an?
Die AMIGA-DOS-News, das Käseblättchen für gestreßte Amiganer, ist nicht ganz so ernstzunehmen. Ernst jedoch war die Erstellung, es gehört schon eine Menge Ausdauer und guter Wille dazu, sich an die Arbeit zu machen.
Die erste Überlegung mußte also dem Aussehen des Layouts im Heft gehören. Eine 1:1-Übernahme war nicht möglich, da dies über den normalen Schriftrand herausgehen würde, eine Verkleinerung ist aber auch nur bis zu einem gewissen Grade möglich, ohne den Leser zu überfordern. Die Texte wurden mit dem in PageSetter integrierten Editor geschrieben, damit auch Steuersequenzen direkt eingebunden werden konnten.
Das vierspaltige Layout fand sein Vorbild im Magazin, gerade bei mehrspaltigen Seiten läßt sich gut ein 'Boulevard-Blatt-Effekt' einstellen. Die Bilder wurden der beiliegenden Artist-Choice-Disk entnommen, hier findet wohl jeder etwas nach seinem Geschmack. Überschriften wurden mit dem Grafik-Editor und den vorhandenen Zeichensätzen erstellt.
Alles in allem dauerte die Erstellung der Seite zirka 12 Stunden (einschließlich zweier Komplett-Abstürze nach Dateizugriff). Ergo: Wer gestalten will, sollte sich Zeit nehmen.

AMIGA DOS Blitzlicht

Name: The Publisher
Art: DTP-Programm-Paket
Hersteller: Gold-Disk
Vertrieb: Rushware....
Preis: 299,– DM

Positiv:
- ausführliches deutsches Handbuch
- Bilderdiskette beiliegend
- in PostScript-Sprache konvertierbar
- externe Textverarbeitung (KindWords) auch normal verwendbar
- leicht zu erlernen

Negativ:
- schlechte Druckauflösung der Fonts auf Matrixdrucken
- Maßangaben nur Inch oder Pica
- bei kleinster Darstellung ungenaues Positionieren der Boxen
- Verschwimmen des Textes bei Unterlegung



Virus-Alarm

bei AMIGA DOS

Wie aus gut unterrichteten Kreisen zu hören ist, hat sich auf den Rechnern der AMIGA DOS- Redaktion ein ganz fürchterlicher Virus eingeschlichen, der den Namen "Killerbuggy" trägt. Besondere Merkmale des Virus sind seine leicht eiförmige Struktur, sowie die insgesamt vier CheckDevice-Routinen, die sich als Endlosschleifen an den eigentlichen Programmkörper anhängen (siehe Bild). Die Funktion dieser relativ neuen Viren ist unbekannt, jedoch gaben Einblicke in das Fast- RAM des Amiga eine Teilfunktion bekannt: Der Virus läuft und läuft und läuft und...

Virus- Killerprogramme wie das "Fetzaway" von Knödelsoft (Kommentar:"'Fetzaway' fetzt alles away!!") machten ihrem Namen alle Ehre, sie fetzten sich selbst aus dem Speicher, jedoch blieben die Killerbuggies unbehelligt, da sie auf ihren vier Endlosschleifen schnell in sämtlichen Devices (DEVS:) verschwanden. Inzwischen soll eine modifizierte Sorte dieses ekligen Viehzeugs gesichtet worden sein, die extrem klein, dafür aber extrem laut agiert, die sogenannten "Trabi-Bandits"! Sie wurden gleich scharenweise im Chip-RAM gesichtet. Verbreitung auf vorhandenes Fast- RAM konnte allerdings noch nicht bemerkt werden. Wir bleiben am Ball!
(H.v.T.)

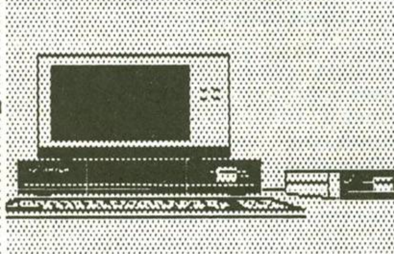
AMIGA DOS - News

unabhängig · überparteilich · überflüssig

Jagdszenen aus Osthessen

Debugging von Assemblerprogrammen





Es gibt drei bekannte Arten, seinen Intellekt und seine Denkfähigkeit an den Rand des Ruins zu bringen: Die erste Art ist die, einen Tag vor Heiligabend die Geschenke für die Verwandtschaft einzukaufen, die Zweite besteht aus der Unmöglichkeit, einem italienischem Kellner beizubringen, daß man absolut keine Nudeln mag, und die Dritte schließlich ist das 'debuggen' von Assemblerprogrammen.

Die Schwierigkeit, auf Anhieb ein perfektes Programm zu schreiben, scheitert zumeist an mehreren Faktoren: Entweder findet irgendeine, im näheren Familienumfeld wichtige Person gerade die neueste CD nicht, der Haushund muß mal gerade ganz dringend aufs oder ans Bäumchen, oder HR3 dröhnt aus Susi's Zimmer, daß sich die Zimmermannsbalken verknoten.

Also kommt man irgendwann an den Punkt, wo der Rechner 'Goodbye' sagt, die Geduld am Ende ist, und der Kaffee zur Neige geht. Wie jedoch beginnt man sinnvollerweise eine 'Anti-Guru-Meditation'?

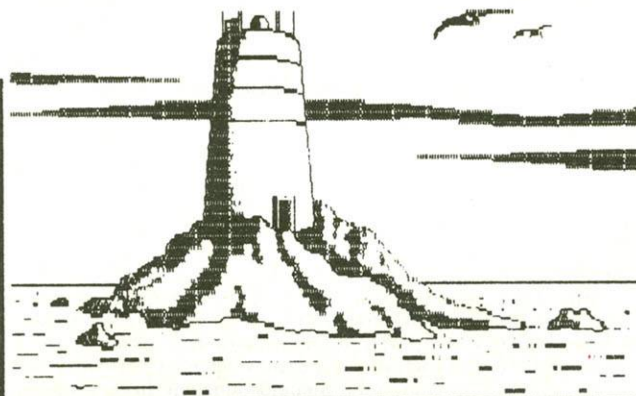
Zuerst lohnt sich eie völlig entspannte Lage vor der Tastatur, am besten in einem auf den Kopf gestellten Schneidersitz (Achtung, keine Schummelei, nicht die Arme benutzen). Räucherstäbchen sind allerdings verpönt (auch die mit Filter), die sechs Sinne sollten möglichst frei bleiben.

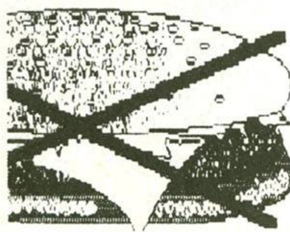
Ein Erfrischungstrunk (empfohlen ist der Schottische) sollte allerdings erlaubt sein. Das Zimmer sollte dabei nicht abgeschlossen werden, damit eventuelle Rettungsaktionen nicht im Sande verlaufen.

So vorbereitet begibt man sich am Besten mit einem Speichermonitor, einem Disassembler, einem One-Step-Simulator und einem ausgefüllten Lottoschein auf die Suche nach dem Bug.

Überfall auf Redaktion

In den späten Abendstunden des 2. Januar 1990 überfielen zwei Maskierte die Redaktion der AMIGA DOS und verlangten lautstark und unter dem Schwenken von PC-Software das sofortige Beenden eines Testes über Musik- Software auf dem Amiga. Es häufen sich die Indizien dafür, daß es sich ebenfalls um im DMV ansässige (und aufsässige) Redakteure einer PC- Zeitschrift handelte, zumal einer der Maskierten unablässig den Satz rief "Hört auf mit dem Sound, ich will es nur noch piepsen hören!!" Erst nachdem den beiden Gangstern das Wort "MS-DOS" zweihundertmal vorgesagt wurde, beruhigte sich die Lage weitgehend, und nach der Beteuerung, daß nur der PC der einzig wahre Computer sei (eine glatte Lüge, die jedoch der Situation angemessen schien), verschwanden die beiden unter lauten Schluchzen in ihren Räumen. Die Redaktion der AMIGA DOS erwägt seit dem die Errichtung einer PC- und MS- DOS- freien Zone!





MARK II SOUND SYSTEM

Der Soundeditor für alle Fälle?

Soft- und Hardware zur Musikerzeugung ist immer mehr im Kommen. Kein Wunder, denn der Amiga ist eine Maschine, die prädestiniert ist für die Musikerstellung. Dabei wachsen in letzter Zeit sogenannte Sound- und Musikeditoren wie Pilze aus dem Boden. MIDI-fähig müssen sie schon sein, und damit das Auge auch noch etwas hat, werden Equalizer mit grafischer LED-Anzeige implementiert. Ja, und wer sowieso professionellen Einsatz genießen will, der benötigt neben der Software auch die entsprechende Hardware. Dabei entstehen natürlich Kosten, die stellenweise horrende Summen annehmen können. Wir haben einen Sound- bzw. Musikeditor unter die Lupe genommen, dessen Name schon seit einiger Zeit in der Gerüchteküche umhergeisterte und der im unteren Preisniveau liegt.

Besonders die vielen Optionen, mit denen Mark II aufwarten kann, machen die Benutzung dieses Editors nicht ganz so einfach. Nach einiger Eingewöhnungszeit wird man jedoch die vielen Möglichkeiten von Mark II nicht mehr missen wollen.

Fangen wir am besten ganz vorn an: Da die Software, bestehend aus zwei Disketten, nicht kopiergeschützt ist, sollte man sich vor der Benutzung ein Sicherheits-Backup

anlegen. Dabei ist es angenehm, daß der Mark II auch Festplatten unterstützt; lange Wartezeiten durch das Laden von Diskette entfallen somit.
Nach einem kleinen Boot-Intro, in dem die Fähigkeiten des Mark II angehört werden können, befindet man sich im Hauptmenü, dem wichtigsten Menü des Soundeditors. Hier lassen sich Songs einfach per Maus einladen und abspeichern. Dabei kommen wir aber schon auf eine Schwäche dieses Produktes zu sprechen, nämlich der Kompatibilität zu anderen Soundeditoren: Mark II kann leider nur sein eigenes Format lesen; Musikstücke aus anderen Editoren werden einfach ignoriert.
Dem Softwarepaket sind sieben Demosongs sowie eine Diskette voller Instrumente beigefügt, die man sich durchaus einmal anhören sollte, bevor man sich an das "Selbstkomponieren" macht. Nach der Auswahl eines Songs werden die zugehörigen Instrumente in den Speicher geladen. Hierbei eine kleine Besonderheit des Mark II: es wird teilweise auch das Fast-RAM unterstützt.
Bevor man an das Editieren geht, sollte man die Taktgröße festlegen. Hierbei ist eine Notenauflösung von maximal 80 bzw. minimal 12 Noten pro Takt möglich. Weiterhin kann angegeben werden, über welchen Kanal bestimmte Takte laufen sollen. So lassen sich 255 voneinander unabhängige Takte programmieren.
Im Instruments-Menü lassen sich bis zu 52 verschiedene Instrumente gleichzeitig einladen, wenn sie im IFF-oder RAW-Format vorliegen. Der Mark II erlaubt es dabei, mit mehreren Instruments-Disketten zu arbeiten, wenn sie den Namen Instruments-, gefolgt von einem alphabetischen Buchstaben, besitzen. Dies hat die Ursache, da der Mark II beim Laden von Instrumenten nach einem bestimmten Schema vorgeht. Zuerst wird beispielsweise die Instruments-a abgefragt, dann Instruments-b etc., bis der Rechner das gesuchte Instrument gefunden hat. Der Vorteil, der daraus resultiert, ist der, daß die Suche nach bestimmten Instrumenten nicht mehr vom Benutzer ausgeht (und wer kennt das nicht, wenn man 20 verschiedene Sample-Disks besitzt...).

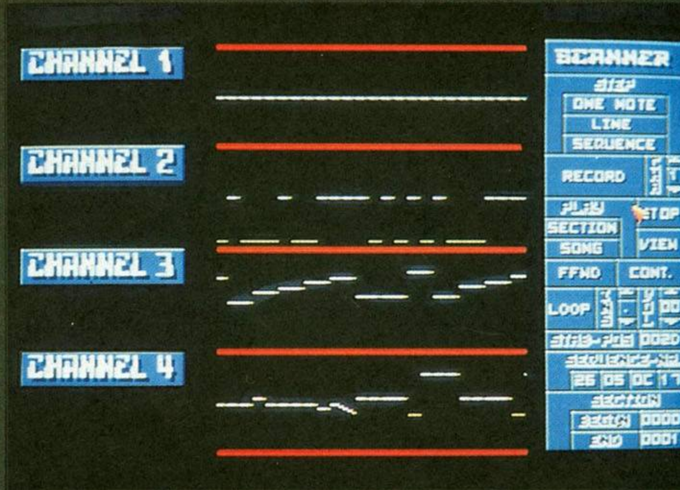


Bild 3: Der Scanner von Mark II Sound System besitzt unter anderem die Aufgabe eines Debuggers, soll also die Fehlersuche vereinfachen

In dem eigentlichen Editor, dem Menü-Arrangement, fallen auf den ersten Blick viele kleine Schalter auf. Wer jedoch denkt, diese lassen sich alle bequem mit der Maus anwählen, der irrt. Leider sind bestimmte Funktionsaufrufe nur über die Tastatur ansteuerbar.
Blockoperationen (cut/paste), wie man sie vom Amiga-Basic oder Textverarbeitungen her kennt, sind in diesem Menü verwirklicht. Dies ist von Vorteil, wenn man bei der Eingabe das Musikstück nicht immer wieder von Anfang an abspielen oder Soundsequenzen kopieren möchte. Beim Kopieren von Blöcken fällt wiederum ein kleines Manko auf. Der zu kopierende Bereich schreibt sich über schon bestehende Daten, anstatt den Block automatisch einzufügen.
Wie bei einem Kassettenrecorder kann man sich die FFWD-Funktion vorstellen: Das laufende Musikstück wird dabei einfach vorgespult.
Das Untermenü Sequence ist unter anderem für die direkte Noten- und Takteingabe, die Instrumentenauswahl und auch die Lautstärke der einzelnen Kanäle zuständig. Dabei wird zuerst das jeweilige Instrument bestimmt und dann die Notenhöhe, bevor die Aussage über den Ton gegeben wird. Die Lautstärke kann für jeden einzelnen Ton gesetzt werden, oder es kann bestimmt werden, ob der Ton ein Portamento, also das Hinüberschweifen zu einem anderen Ton, erfahren soll. Damit lassen sich die ASDR-Werte (Hüllkurven) leicht manipulieren. Falls man kein Portamento aktivieren möchte, kann auch durch sogenanntes Swapping eine Note mit der des benachbarten Kanals (Kanal 1 und Kanal 2 bzw. Kanal 3 und Kanal 4) ausgetauscht werden. Das Tauschen der Noten zwischen Kanal zwei und drei ist zwar auch möglich, jedoch ginge dadurch der Stereoeffekt verloren.
Zur schnelleren Noteneingabe dient der Extended Edit. Dabei kann eine gesamte Reihe, also alle vier Kanäle gleichzeitig, editiert werden. Die Eingabe in der ersten Spalte wird dann gleich auf die anderen drei Spalten übertragen.

Der Scanner – Fehlersuche einmal anders

Im Scanner-Menü werden die einzelnen Töne grafisch dargestellt. Dabei entspricht die Höhe der Punktlage der Höhe der jeweiligen Note. Dies erleichtert die Suche nach falschen Tönen. Hier kann das Musikstück noten- oder sequenzweise, aber auch als gesamtes Lied abgefahren werden, der Bildschirm wird dabei von rechts nach links gescrollt.
Der Record-Modus dient zur Einspielung von Melodien über die Tastatur in Echtzeit. Hierbei kann man sich das Instrument und die Lautstärke vorher einstellen. Nach Auswahl des Kanals, auf dem die Vorlage gespielt werden soll, kann man diesen starten und dann die Begleitung dazu über die Tastatur einspielen.
Die Geschwindigkeit läßt sich bei dieser Option zudem noch etwas manipulieren.
Der Mark II besitzt ein Klaviaturmenü, mit dem man das Keyboard wie ein Klavier benutzen kann. Will man beispielsweise ein im Speicher befindliches Musikstück begleiten, so ist das in diesem Menü ermöglicht. Es werden nur die beiden ersten Kanäle gespielt, die (zweistimmige) Begleitung übernimmt dann der Benutzer auf den anderen Kanälen. Dies ermöglicht die aktive Teilnahme an einer laufenden Melodie, wobei es an dem Benutzer ist, welches Instrument er spielen möchte.

Abspeichern auf Diskette – aber stand-alone bitte!

Will man ein erstelltes Musikstück abspeichern, so kann man es auch mit allen geladenen Instrumenten und mit einer Abspielroutine versehen, wodurch diese Module eigenständig lauffähig werden.
Ziehen wir nun den Schlußstrich: Der Mark II ist ein Musik- und Soundeditor, der für wenig Geld viel leistet. Zwar ist die Bedienung etwas kompliziert, aber nach etwas Einarbeitungszeit ist auch diese Hürde zu meistern. Wer gerne auf seinem Amiga komplexe Musikstücke komponiert, ohne großen Wert auf MIDI-Fähigkeit zu legen, der sollte sich dieses Produkt auf jeden Fall einmal ansehen.

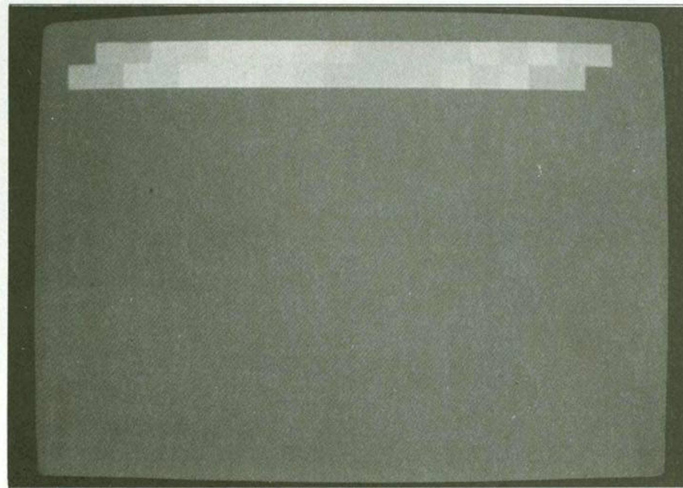
(br)

AMIGA DOS Blitzlicht

Name: Mark II Sound System
Hersteller: Cachet
Quelle: Cachet
Ostendstr. 32
7524 Oestringen
Preis: ca. 80 DM

Positiv:
- geringer Preis
- Takte unabhängig voneinander programmierbar
- Recordmodus
- 52 Instrumente gleichzeitig verfügbar

Negativ:
- es können nur Mark-II-Sounds geladen werden
- keine MIDI-Unterstützung



Ulf Peterson

Blitter-Ecke (Teil 2)

Kopieren und Animationen

Wie im ersten Teil unseres Blitterkurses angekündigt, wollen wir Ihnen diesmal zeigen, wie man mit dem Blitter Daten kopiert und mit ihm auch bequem Animationen erzeugen kann. Der gleich folgende Text gilt als Einstieg in die wunderbare Welt des Blitters und sollte von Ihnen deshalb aufmerksam gelesen werden, da er die Grundlage für die nächsten Teile, insbesondere den, der den BOBs gewidmet ist, bildet. Falls es erforderlich sein sollte, lesen Sie sich am besten den Artikel noch ein zweites Mal durch oder senden uns Ihre Fragen schriftlich zu (Rückporto).

Wie im ersten Teil dieses Kurses schon erwähnt wurde, stellt der Blitter drei Quellbereiche und einen Zielbereich, der für die fertige Datenausgabe zuständig ist, dem Anwender zur Verfügung. Dabei sind sowohl die Quellen, als auch der Datenzielbereich genauso wie eine normale Bitplane aufgebaut, also in Spalten und Zeilen unterteilt. Aus diesem Grund eignet sich der Blitter auch hervorragend zur Bearbeitung von Grafiken bzw. Bitplanes, durch die sich der AMIGA im besonderen auszeichnet. Allerdings reicht es dem Blitter nicht, wenn man ihm nur die Adressen der verschiedenen Bereiche angibt, sondern man muß ihm schon einige weitere, wichtige Mitteilungen über die benutzten Abschnitte machen. So ist es unter anderem unbedingt erforderlich anzugeben, wie groß eigentlich der Bereich ist, der vom Blitter bearbeitet werden soll.

Grundlagen

Dieser Bereich, der in Fachkreisen als Blitterfenster bezeichnet wird, stellt sozusagen den Arbeitstisch des Blitters dar. Im Klartext bedeutet dies, daß der Blitter nur die Ausschnitte eines oder mehrerer Bereiche bearbeitet, die der Größe des Blitterfensters entsprechen. Schauen Sie sich zum besseren Verständnis am besten die Zeichnung 1 an, in der alles das noch einmal zeichnerisch erläutert wird, was wir eben versucht haben, in Worten darzulegen. Neben der Größenangabe dient das Blitterfenster gleichzeitig auch noch als der Aktivierer einer Blitteroperation. Aus diesem Grunde steht diese Anweisung auch immer am Ende einer Blitterfunktion, die normalerweise so aufgebaut ist:

○ Auswählen der verwendeten Quellbereiche und des Zielbereichs. Anschließend deren Adressen im Speicher (CHIP-RAM!) angeben.

○ Festlegung der zu beachtenden Verknüpfungen.

○ Verschiebungen, Maskierungen und die Adressierungsrichtung anwählen.

○ Mit der Festlegung des Blitterfensters die Blitteroperation starten.

(Tabelle 1 zeigt die Festlegung eines Blitterfensters.) Eine einfache Formel steht für die Beschreibung dieses Registers zur Verfügung:

BLTSIZE = Höhe*64 + Breite (in Worten !)

Modulo

Wie Sie anhand der Zeichnung vielleicht schon festgestellt haben, entspricht die Größe des Blitterfensters nicht immer auch der der verwendeten Bereiche. Damit der Blitter die Daten aber trotzdem einwandfrei kopiert, muß man ihm die Differenz der Breite des Blitterfensters und des Grafik- oder Zielbereiches in Bytes mitteilen. Wie bei den normalen Bitplanes sind auch beim Blitter dafür sogenannte Modulo-Register zuständig. Sobald der Blitter eine Zeile mit der Breite des Blitterfensters abgearbeitet hat, werden die Inhalte der verschiedenen Modulo-Register zu den Adressen der verwendeten Bereiche addiert. Auf diese Weise werden die nicht verwendeten Teile eines Bereiches übersprungen, und man hat die Garantie, daß alles korrekt kopiert wird und beispielsweise später kein Grafiksalat entsteht, der bei Nichtbeachtung der Modulos entstehen würde.

Ascending/Descending Mode

Normalerweise wählt man für Grafiken immer eigene Bereiche, so daß keine übergreifenden Ansprüche seitens der Bitplanes entstehen. Es gibt aber mitunter auch Fälle, wo dem nicht so ist und sich die Bereiche teilweise überlappen. Schaubild Nummer 2 verdeutlicht die zwei typischen Situationen, die beim Kopieren von Daten auftreten

Register: BLTSIZE $DFF058 Dient der Festlegung des Blitterfensters

Aufbau:

Bit-Nr.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bezeichnung	H9	H8	H7	H6	H5	H4	H3	H2	H1	H0	B5	B4	B3	B2	B1	B0

H9-H0: Durch diese zehn Bits gibt man an, wie hoch das Blitterfenster ist, also wie viele Zeilen es umfaßt. Die maximale Höhe kann dabei 1024 Zeilen betragen, wobei die Zeile 1024 durch das Nullsetzen aller Hx-Bits angegeben wird.

B5-B0: Diese Bits geben die Breite des Fensters in Worten (!!!) an. Oftmals wird diese Eigenschaft übrigens vergessen, und es entstehen mitunter kuriose Gebilde bei falscher Angabe. Die maximale Breite kann 64 Worte betragen, wobei die 64 wiederum durch das Nullsetzen aller zuständigen Bits erreicht wird.

(Tabelle 1)

Name	Register	Beschreibung des Registers	
BLTAFWM	$DFF044	Maske für das erste Wort einer Zeile	
BLTALWM	$DFF046	Maske für das letzte Wort einer Zeile	
BLTCPTH	$DFF048	Anfangsadresse des	High-Wort (H)
BLTCPTL	$DFF04A	Quelldatenbereiches C	Low-Wort (L)
BLTBPTH	$DFF04C	Anfangsadresse des	H
BLTBPTL	$DFF04E	Quelldatenbereiches B	L
BLTAPTH	$DFF050	Anfangsadresse des	H
BLTAPTL	$DFF052	Quelldatenbereiches A	L
BLTDPTH	$DFF054	Anfangsadresse des	H
BLTDPTL	$DFF056	Zieldatenbereiches D	L
BLTCMOD	$DFF060	Quelle C Modulo-Wert	
BLTBMOD	$DFF062	Quelle B Modulo-Wert	
BLTAMOD	$DFF064	Quelle A Modulo-Wert	
BLTDMOD	$DFF066	Zielbereich D Modulo-Wert	

(Tabelle 2)

können. Konzentrieren wir uns dabei auf die Abbildung A, die den eben angesprochenen Sonderfall verdeutlicht. Wenn in diesem Fall der normale Kopiermodus verwendet wird (Ascending Mode), würden falsche Daten übertragen werden, denn Quelle A überschreibt sich wegen der Überlappung ja selbst, noch bevor diese deren echten Endbereich übertragen hat. Um diese Situation zu vermeiden, bietet der Blitter noch zusätzlich den sogenannten Descending Mode, bei dem, wie der Name es schon andeutet, alles umgekehrt abläuft. Dementsprechend darf man dem Blitter auch nicht die Anfangsadressen der verwendeten Bereiche angeben, sondern deren Endadressen, da der Blitter jetzt von unten nach oben arbeitet. Auf diese Weise hätten wir dann das Problem der Abbildung A gelöst. Es sei erwähnt, daß der Descending Mode relativ selten verwendet wird, aber der Vollständigkeit halber wollten wir ihn hier nicht unerwähnt lassen.

Maskierung

Es kann relativ häufig vorkommen, daß man Grafiken oder ganz einfach Daten kopieren will, deren Anfangs- und Endbyte nicht genau auf einem Wortende bzw. Wortanfang liegen. Nun wäre es nicht sehr erfreulich, wenn man mit den eigentlichen, gewünschten Daten auch noch überflüssige Datenbits mitkopiert. Aus diesem Grunde bietet der Blitter die Möglichkeit, die Randbereiche zu maskieren, um unerwünschte Bits herauszufiltern. Mit Randbereichen sind dabei das erste und das letzte Datenwort einer Zeile gemeint. Leider existiert die Möglichkeit des Ausfiltrierens nur für die Quelle A. Wenn Sie also eine Grafik mit mehreren Bitplanes kopieren wollen, die nicht genau auf einer Wortgrenze liegt und keine überflüssigen Bits enthalten soll, müssen Sie dies mit mehrmaliger Verwendung der Quelle A tun.

Wie kann man aber nun die Maskierung einstellen?

Dazu schreiben Sie ganz einfach in die betreffenden Maskenregister ein Wort, in dem alle Bits nicht gesetzt sind, wo eine Ausfiltrierung, also Auslöschung erfolgen soll. Wenn also zum Beispiel die beiden ersten Bits einer Zeile unberücksichtigt bleiben sollen, müssen die beiden ersten Wortbits der Maske ungesetzt sein (Binär: 0011111111111111). Damit wäre auch dieser Abschnitt hinreichend erklärt. Es folgt eine Übersicht der bisher verwendeten Register. (Siehe Tabelle 2)

Verschieben der Bereiche

Mitunter kann es vorkommen, daß man einen bestimmten Teil eines Datenbereiches nur um einige Bits im Bereich selbst nach rechts verschieben oder einfach nur an eine Position kopieren will, die nicht genau auf einer Wortgrenze liegt. Aus diesem Grund können die Daten der Bereiche A und B um bis zu 15 Bits nach rechts verschoben werden, wenn es erforderlich ist. Dabei werden alle Bits, sofern sie über eine Wortgrenze hinausgeschoben werden, in das nächstgelegene Wort verschoben. Es gehen also keine Bits in irgendeiner Weise verloren. Eine derartige Verschiebung von Daten kostet den Blitter übrigens keine zusätzliche Zeit.

Logische Verknüpfungen

Kommen wir zu einem sehr wichtigen Abschnitt dieses Blitterkurses, den Verknüpfungen. Mit diesen ist es möglich, dem Blitter vorzuschreiben, bei welchen Kombinationen der Eingangsbits, also der der Quellen, das Zielbit des Zielbereiches D gesetzt sein soll bzw. sein kann. Bei Benutzung dieses Systems ist übrigens unbedingt darauf zu achten, unbenutzte Quellen mit Hilfe der für Verknüpfungen zuständigen LFx-Bits zu neutralisieren. Dazu ein einfaches Beispiel. Wir wollen eine kleine Grafik in eine Bitplane kopieren. Mathematisch ausgedrückt (Boolesche Algebra) müßte man dafür schreiben: A = D. (Wenn ein Bit in der Quellgrafik gesetzt ist, soll es auch in der Zielgrafik gesetzt sein. Es wird also der Bereich originalgetreu kopiert. a = D hätte genau den umgekehrten Effekt zur Folge, denn dann würde man die Quellgrafik invertiert kopieren.)

Wir können jetzt aber nicht nur z.B. das LFx-Bit Nr.7 nehmen, um diesen Ausdruck festzulegen. Da die Quellen B und C nicht verwendet werden, müssen wir sie bei der Verknüpfung ausschalten. Mit anderen Worten heißt das, wir müssen alle LFx-Kombinationen wählen, bei denen A gesetzt ist. Dabei erreichen wir automatisch, daß b und B, c und C in den Kombinationen vorkommen. Da dies immer eine positive Ausgabe ergibt, haben wir diese Bits der Quellen B und C. ausgeschaltet. Der Blitter muß nur noch auf den Zustand des Bits der Quelle A achten.

Wer sich mit der Booleschen Algebra vom Mathematikunterricht her auskennt, kann sich auch damit die zuständigen LFx-Bits errechnen. Dazu das oben genannte Beispiel:

A = D A*(b + B)*(c + C) = D
A*(bc + bC + Bc + BC) = D
Abc + AbC + ABc + ABC = D

Jetzt muß man nur noch diese errechneten LFx-Bits im BLTCON0-Register setzen. (siehe Tabelle 3)

Bit-Nr.	Bezeichnung	Erläuterung
Register: BLTCON0 $DFF040		
15	ASH3	Die Bits 12-15 enthalten den Wert
14	ASH2	für die Verschiebung der Datenquelle A
13	ASH1	an Bildschirmpunkten
12	ASH0	
11	USEA	Quelle A soll benutzt werden
10	USEB	Quelle B soll benutzt werden
9	USEC	Quelle C soll benutzt werden
8	USED	Ziel D soll benutzt werden (ist immer der Fall)
7	LF7	Miniterm ABC benutzen (Bitkombination 111)
6	LF6	Miniterm ABc benutzen (110)
5	LF5	Miniterm AbC benutzen (101)
4	LF4	Miniterm Abc benutzen (100)
3	LF3	Miniterm aBC benutzen (011)
2	LF2	Miniterm aBc benutzen (010)
1	LF1	Miniterm abC benutzen (001)
0	LF0	Miniterm abc benutzen (000)
Register: BLTCON1 $DFF042		
Bit-Nr.	**Bezeichnung**	**Erläuterung**
15	BSH3	Die Bits 12-15 enthalten den Wert
14	BSH2	für die Verschiebung der Datenquelle B
13	BSH1	an Bildschirmpunkten
12	BSH0	
13-5		unbenutzt
4	EFE	Exclusive Fill Enable
3	IFE	Inclusive Fill Enable
2	FCI	Fill Carry In
1	DESC	Descending Mode ein- oder ausschalten
0	LINE	Linienmodus ein- oder ausschalten

(Tabelle 3)

Animation

Übersetzt bedeutet Animation ja im Grunde genommen, daß eine Grafik bewegt wird oder der Effekt einer Bewegung entsteht. Je besser und vielfältiger dabei eine Animation ist, umso realitätsnäher und fließender ist sie auch. Nun bietet sich für Animation der Blitter ja geradezu an, da er Daten mit einer unglaublichen Geschwindigkeit kopieren kann. Wenn Sie also eine animierte Grafiksequenz erstellt haben, kopieren Sie mit Hilfe des Blitters die einzelnen Sequenzen der Animation in die gewünschte Grafik mit einer bestimmten Zeitsequenz. Des weiteren ist zu beachten, daß der Kopiervorgang möglichst in der vertikalen Austastlücke stattfinden sollte, da sonst sehr häßliche Flacker- oder Resterscheinungen auftreten können. Abbildung drei verdeutlicht das Thema Animation mit einer kurzen Schautafel.

Damit wäre auch der zweite Teil des Blitterkurses beendet. Das nächste Mal wird er für viele noch interessanter werden, denn dann werden wir auf das Thema BOBs eingehen. Bis dahin noch viel Spaß mit dem Blitter und dem abgedruckten, einfach aufgebauten Beispiellisting.

(mm)

Listings



```
                        opt 1-

;************************************
;*BLITTER-KURS fuer AMIGADOS Teil 2*
;*      1989,90 by Ulf Petersen     *
;*  Listing erstellt mit DEVPAC II *
;************************************

BLTAPTH         equ     $50         ;Adresse der Quelle A
BLTDPTH         equ     $54         ;"  "     des Zielberei
ches D
BLTAMOD         equ     $64         ;Modulo- Wert fuer Dat
en-
BLTDMOD         equ     $66         ;quelle A und Ziel D
BLTCON0         equ     $40         ;Kontrollregister 1
BLTCON1         equ     $42         ;Kontrollregister 2
BLTSIZE         equ     $58         ;Groesse des Blitterfe
nsters
BLTAFWM         equ     $44         ;Maskierung des linken
 Randes
BLTALWM         equ     $46         ;    "       "  rechte
n Randes

COP1LC          equ     $80         ;Erste Copper-Liste
COP2LC          equ     $84         ;Zweite    "
COPJMP1         equ     $88         ;Initialisierungsregis
ter Liste 1
COPJMP2         equ     $8a         ;        "
      "    2

DIWSTRT         equ     $8e         ;Anfang des Bildschirm
fensters
DIWSTOP         equ     $90         ;Ende des Bildschirmfe
nsters
DDFSTRT         equ     $92         ;Bitplane-DMA Anfang
DDFSTOP         equ     $94         ;Bitplane-DMA Ende
BPL1PTH         equ     $e0         ;Adresse der ersten Bi
tplane
BPL1PTL         equ     $e2         ;        "
BPLCON0         equ     $100        ;Kontrollregister der
Bitplane
BPLCON1         equ     $102        ;Scrolling
BPLCON2         equ     $104        ;Prioritaeten Grafik/S
prites
BPL1MOD         equ     $108        ;Modulo-Wert ungerader
Bitplanes
BPL2MOD         equ     $10a        ;    "      gerader
Bitplanes
SPR0PTH         equ     $120        ;Adressen der Spriteta
bellen
SPR0PTL         equ     $122

CIAAPRA         equ     $BFE001
DMACON          equ     $96         ;DMA-Kontrollregister
(write)
DMACONR         equ     2           ;        "
(read)
STARTLIST       equ     38
Random          equ     6

; Library-Adressen

Forbid          equ     -30-102
Permit          equ     -30-108
AllocMem        equ     -30-168
FreeMem         equ     -30-180
OpenLibrary     equ     -30-522

Exec            equ     4
ClearChip       equ     2+$10000

Size1           equ     40*256      ;Groesse der Bitplane
Size2           equ     100         ;   "     "  Copper-Li
ste
Size3           equ     12*32       ;   "     "  Grafik A

; *** Start des Programms ***

START           move.l  Exec,a6

; Speicherplatz fuer die Bitplane anfordern ...

                move.l  #Size1,d0
                move.l  #ClearChip,d1
                jsr     AllocMem(a6)
                move.l  d0,Bitplane

; Speicherplatz fuer die Copper-Liste anfordern ...

                move.l  #Size2,d0
                move.l  #ClearChip,d1
                jsr     AllocMem(a6)
                move.l  d0,Liste

; Speicher fuer die Grafik anfordern ...

                move.l  #Size3,d0
                move.l  #ClearChip,d1
                jsr     AllocMem(a6)
                move.l  d0,Grafik

; Copper-Liste direkt erstellen ...

                move.l  Liste,a0
                move.l  Bitplane,d0

; Bitplaneadresse in die Copper-Liste eintragen

                move.w  #BPL1PTL,(a0)+
                move.w  d0,(a0)+
                swap    d0
                move.w  #BPL1PTH,(a0)+
                move.w  d0,(a0)+

; Alle Sprites auf leere Datentabellen umlenken ...

                move.l  Liste,d0
                add.l   #96,d0
                move.l  #SPR0PTL,d1
                move.l  #SPR0PTH,d2
                moveq   #7,d3

Next            move.w  d1,(a0)+
                move.w  d0,(a0)+
                add.l   #4,d1
                swap    d0
                move.w  d2,(a0)+
                move.w  d0,(a0)+
                add.l   #4,d2
                swap    d0
                dbf     d3,Next

                move.l  #$fffffffe,(a0)     ;Ende der Cop
per-Liste

                lea     $dff000,a5

                move.l  Liste,COP1LC(a5)    ;Aktivierung
der neuen
                clr.w   COPJMP1(a5)         ;Copper-Liste

                move.w  #$3081,DIWSTRT(a5)  ;Standardbild
schirm auf-
                move.w  #$30c1,DIWSTOP(a5)  ;bauen mit ei
ner
                move.w  #$0038,DDFSTRT(a5)  ;Bitplane.
                move.w  #$00d0,DDFSTOP(a5)
                move.w  #%0001001000000000,BPLCON0(a5)
                clr.w   BPLCON1(a5)
                clr.w   BPLCON2(a5)

                move.w  #$83c0,DMACON(a5)

; Der folgende Programmteil fuellt den Quelldatenbereich
 A mit
; einem Kastenmuster.

                move.l  Grafik,a0
                move.l  #%000000000000000011111111111111
11,d0
                move.l  #3*16-1,d1
Next2           move.l  d0,(a0)+
                  dbf   d1,Next2

                move.l  #%111111111111111100000000000000
00,d0
                move.l  #3*16-1,d1
Next3           move.l  d0,(a0)+
                  dbf   d1,Next3

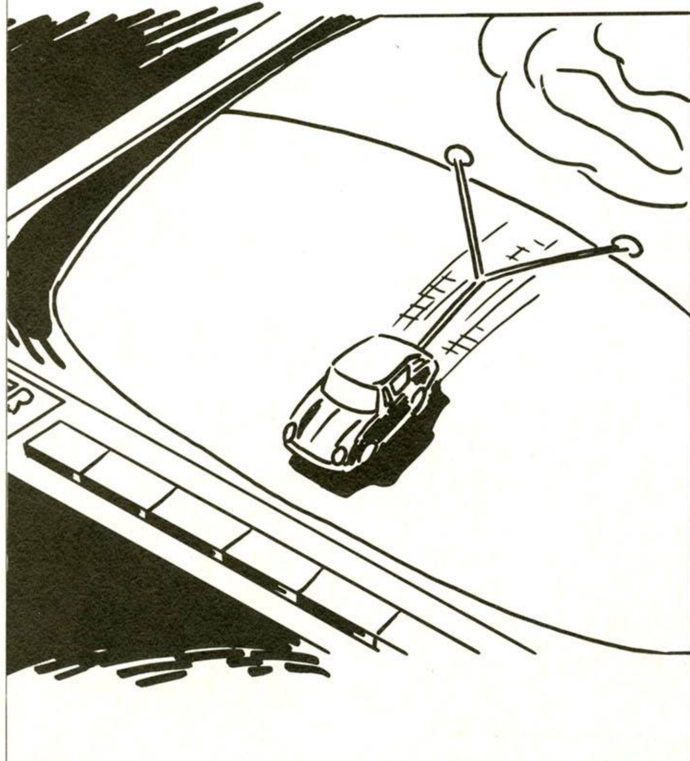
; Hauptprogramm: Die am Anfang stehende Warteschleife ka
nn
; ausgelassen werden, um den Blitter in Echtzeit arbeite
n zu
; lassen. Loeschen Sie dazu bloss "bne.s      Go_on".

                clr.l   d1
Go_on           move.l  $dff004,d2          ;Bildschirm e
                and.l   #$1ff00,d2          ;aufgebaut ?
                cmp.l   #$2b00,d2
                bne.s   Go_on               ;Nein, also w
arten !

; Es folgt die Initialisierung der Blitterregister.

                clr.w   BLTAMOD(a5)         ;Kein Quell-
Modulo
                move.w  #28,BLTDMOD(a5)     ;Ziel- Modulo
-Wert
                move.l  Grafik,BLTAPTH(a5)  ;Adresse der
Quellgrafik
                move.l  Bitplane,d0         ;Adresse der
Bitplane
                add.l   #1,d1               ;um Verschieb
ungswert
                cmp.b   #28,d1              ;erhoehen und
 an d0
                bne.s   Next4               ;uebergeben
                moveq   #0,d1
Next4           add.l   d1,d0
                move.l  d0,BLTDPTH(a5)      ;Endwert der
Zieladresse eintragen
                clr.w   BLTCON1(a5)
                move.w  #$ffff,BLTAFWM(a5)  ;Keine Ausmas
kierung !
                move.w  #$ffff,BLTALWM(a5)
                move.w  #%0000100111110000,BLTCON0(a5)
                move.w  #32*64+6,BLTSIZE(a5) ;Groesse des
 Blitterfensters
```

```
                                                      ;eintragen und damit den
                                                      ;Blitter aktivieren
wait       btst        #14,DMACONR(a5)                ;Warteschleife, bis der
           bne.s       wait                           ;Blitter fertig ist.
           move.w      #40*32-1,d0                    ;Grafikbereich der
           move.l      Bitplane,a0                    ;Bitplane wieder loeschen,
Next5      clr.b       (a0)+                          ;um korrekte Kopierungen
           dbf         d0,Next5                       ;zu erreichen
           btst        #6,CIAAPRA                     ;Linke Maustaste gedrueckt ?
           bne.s       Go_on                          ;Nein,also noch einmal!
; Die am Anfang reservierten Speicherbereiche wieder fuer eine
; weitere Verwendung freigeben.
           move.l      #Size3,d0                      ;Zuerst die Grafik A
           move.l      Grafik,a1
           jsr         FreeMem(a6)
           move.l      #Size2,d0                      ;Copper-Liste
           move.l      Liste,a1
           jsr         FreeMem(a6)
           move.l      #Size1,d0                      ;Bitplane
           move.l      Bitplane,a1
           jsr         FreeMem(a6)
; Alte Copper-Liste wieder aktivieren ...
           lea         GRFX,a1
           clr.l       d0
           jsr         OpenLibrary(a6)
           move.l      d0,a4
           move.l      STARTLIST(a4),COP1LC(a5)
           clr.w       COPJMP1(a5)
           move.w      #$83e0,DMACON(a5)
           rts                                        ;und raus...
; CHIP-RAM Adressen
Bitplane   dc.l        0                              ;BITPLANE
Liste      dc.l        0                              ;COPPER-LISTE
Grafik     dc.l        0                              ;GRAFIK A
GRFX       dc.b        "graphics.library",0
```



Bernd Rudolf

Es lebe die Mathematik!

Funktionen grafisch ausgegeben

Ableitungen, Asymptoten, Intervalle - Bei diesen Worten sträuben sich vielen Schülern sicherlich die Haare. Zur Überprüfung, ob man seine Berechnungen korrekt durchgeführt hat, sollten die Graphen zu den mathematischen Kurven gezeichnet werden. Unser Programm FunkPlot soll diese Aufgabe übernehmen.

Das GFA-BASIC-Programm erstellt die Graphen zu Funktionen in jedem beliebigen Intervallabschnitt.
FunkPlot ist (leider) nicht zur Differential- beziehungsweise Integralrechnung geeignet, sondern kann "nur" Funktionen und die zugehörigen ersten drei Ableitungen zeichnen. Die Frage nach dem Sinn und Zweck von Ableitungen wird immer wieder von geplagten Schülern gestellt. Mit Ableitungen können beispielsweise Hoch-, Tief- und Wendepunkte einer mathematischen Funktion bestimmt werden. Ebenso kann man Aussagen über die Wendestellen und das Monotonieverhalten (Steigungsverhalten) machen (siehe Box).

Monotones Verhalten mathematischer Kurven

Wenn Ihnen diese Aufstellung überhaupt nichts sagt, so lassen Sie sich doch vom Programm einfach einmal die Funktion

($f(x) = x^4-4*x^2 + 3$)

mit ihren ersten zwei Ableitungen zeichnen und vergleichen das Bild mit der Tabelle. So viel zum Hintergrund. Probleme für das Ableiten im allgemeinen sind nicht-stetige Funktionen, das heißt Funktionen, die eine Definitionslücke haben. Dies sind Funktionen wie $f(x) = 1/x$. An der Stelle 0 ist diese nicht definiert. Folglich ergäbe sich beim Zeichnen oder Berechnen des Funktionswertes an der Stelle 0 ein "Division by Zero-Error", der zum Abbruch des Programms führen würde. Um dies zu verhindern, enthalten alle Zeichen- und Rechenroutinen Sicherheitsabfragen. Das gleiche gilt für den Zeichenbereich. GFA-Basic "stürzt" zwar nicht ab, falls die Koordinaten eines Punktes im negativen Bereich liegen, es kann jedoch zu einem Absturz kommen, wenn vor oder hinter dem Zeichenbildschirm andere wichtige Daten liegen, die dann überschrieben werden.

Doch nun zum eigentlichen Programm: Das erste Problem, das sich stellt, ist das Definieren der Funktionen. Das einfachste wäre, ein String einzulesen, das man hinterher irgendwie in eine Funktion umwandelt.

Dafür gab (und gibt) es beim 'guten, alten' C-64 die Möglichkeit des programmierten Direktmodus, um eine Zeile in das laufende Programm einzufügen. So geht es mit GFA-Basic aber leider nicht. Nicht einmal ein MERGE-Befehl steht zur Verfügung. Außerdem wird das Neu- oder Umdefinieren einer mit DEFFN erstellten Funktion mit einer Fehlermeldung geahndet.

Hier hilft nur eines: der Trick mit dem Knick aus Omas Spinatkiste mit dem Blubb! In dem Menü 'Funktionen' sind alle Standardfunktionen aufgeführt, die man für den "Hausgebrauch" benötigt.

Durch Anwählen des Menüpunktes 'Erstellen' erscheint ein Fenster in der Bildschirmmitte. Nun sind alle übrigen Menüpunkte "disabled", also nicht mehr anwählbar. Es lassen sich nur noch die Menüpunkte unter 'Funktionen' auswählen. Um nun eine spezielle Funktion zusammenzustellen, müssen Sie lediglich nacheinander die einzelnen Teilfunktionen wählen, die durch '+' verbunden werden, und den Faktor oder den Exponenten eingeben. Ist die Zusammenstellung komplett, wählen Sie einfach 'Beenden'. Die Faktoren aller Teilfunktionen, die Sie für Ihren Graphen nicht benötigen, sind auf Null gesetzt, haben also auf das Ergebnis keinerlei Auswirkungen. Lediglich die Geschwindigkeit der Zeichenroutine wird verringert.

"Doch was ist, wenn ich beispielsweise den Graphen von f(x) = 1/SQR(ABS(SIN(x^2))) zeichnen will?", wird sich der eine oder andere nun fragen. Dies ist nur möglich, wenn Sie die Funktion direkt in das Programm schreiben. Dafür ist die Funktion 'selbst(x)' vorgesehen. Sie befindet sich ganz am Anfang des Programms. Es ist in diesem Fall unbedingt notwendig, das Programm nicht in der compilierten Version vom CLI, sondern vom Editor aus zu starten. Andernfalls ließe sich die Voreinstellung der selbstdefinierten Funktion nicht ändern. Die Voreinstellung sieht weiterhin vor, daß diese Funktion gezeichnet wird, falls keine andere definiert wurde.

Wurde zwischenzeitlich eine andere Funktion definiert, so läßt sich die Einstellung durch Anwählen des Menüpunktes 'Vordef.' wieder rückgängig machen.

Im Menü 'Layout' sind alle die Grafik betreffenden Unterroutinen aufgeführt. 'Zeichnen' bringt die definierte Funktion auf den Bildschirm. Nach der Wahl von 'Ableiten' und der Eingabe der gewünschten Ableitung (1 bis 3) erhalten Sie den Graphen, der sich je nach Ableitungsgrad in der Farbe unterscheidet. Es steht Ihnen frei, zwischendurch beliebig oft den Bildschirm zu 'löschen', um die Übersichtlichkeit zu erhöhen.

...alles zu Papier bringen? Null Problemo!

Ein kleines Problem könnte sich bei der Routine 'Print' ergeben. Sie geht davon aus, daß der Drucker weiße Flächen weiß und entsprechend schwarze Flächen schwarz druckt. Deshalb muß der gesamte Bildschirm noch einmal in anderen Farben gezeichnet werden. Wem das zu umständlich ist, der kann die Farben bereits im Hauptprogramm auf die Grauwerte einstellen und die gesamte Farbänderung sowie die erneuten Aufrufe der Zeichenroutinen löschen. Das geht natürlich auf Kosten der Augenfreundlichkeit. Leider ist es nicht ohne Aufwand möglich, die Titelleiste des Screens abzuschalten. Erschrecken Sie also nicht, wenn Ihr Drucker in der ersten Zeile mit Getöse loslegt.

Das Menü 'Tools' stellt Ihnen drei Utilities zur Verfügung, die von Nutzen sind, wenn Sie bereits eine oder mehrere Funktionen gezeichnet haben. Dann können Sie beispielsweise mit dem 'Lineal' die Funktionswerte vergleichen, indem Sie die Maus in die richtige Position fahren und dann in der linken oberen Ecke die Koordinaten des Mauszeigers ablesen. Die Koordinaten beziehen sich auf das momentane Koordinatensystem, nicht auf den Bildschirm. Durch Drücken der linken Maustaste kehren Sie

Funktion: f(x)
1. Ableitung: f'(x)
2. Ableitung: f''(x)
3. Ableitung: f'''(x)

Stationäre Stellen:
f'(x) = 0

Extremstellen:
f'(x) = 0 und f''(x) < > 0

Hochpunkte:
f'(x) = 0 und f''(x) < 0
Tiefpunkte:
f'(x) = 0 und f''(x) > 0

Wendepunkte:
f''(x) = 0 und f'''(x) < > 0

Sattelstellen:
f'(x) = 0 und f''(x) = 0 und f'''(x) < > 0

Monotonie:
f'(x) > 0; xEA = > f ist streng monoton steigend im Intervall A.
f'(x) < 0; xEA = > entsprechend fallend.
Ist f'(x) größer- (kleiner-) gleich 0, so ist f nur monoton steigend (fallend).

Box 1

wieder ins Hauptmenü zurück. Durch mehrfaches Wählen des Punktes 'Interv' können Sie die Anzeige des Intervalles an- und ausschalten. Das kleine Fenster, in dem die Intervallabstände angezeigt werden, läßt sich problemlos an jeder beliebigen Stelle des Bildschirms plazieren. So läßt sich eine Störung des Bildes vermeiden.

Mit der Option 'f()' kann der Funktionswert der aktuellen Funktion mit ihren Ableitungen an jeder Stelle (x) ermittelt werden. Dafür brauchen sie nur den Mauszeiger an die entsprechende Ordinate (Stelle auf der x-Achse) zu bewegen und in der linken, oberen Ecke des Bildschirms den Funktionswert abzulesen. Im Gegensatz zum Lineal bezieht sich dieses Hilfsmittel nur auf die momentan definierte Funktion. Mit fortschreitendem Ableitungsgrad müssen leider Ungenauigkeiten der Funktionswerte in Kauf genommen werden, da die Ableitung auf grafischem Weg erfolgt. Hierzu ein kleines Beispiel:
Will man einen Funktionswert der zweiten Ableitung ermitteln, so benötigt man drei Funktionswerte der Ausgangsfunktion. Nun wird die Differenz des ersten und zweiten, dann des zweiten und dritten Funktionswertes berechnet und mit dem negativen Kehrwert der Schrittweite ((f(1/10)-f(2/10))*(-10)) multipliziert.

Do it yourself – 3D-Darstellung

Auf diese Weise erhält man zwei neue Funktionswerte, nämlich die der ersten Ableitung an diesen Stellen. Wenn wir nun dasselbe Verfahren auf diese zwei Funktionswerte anwenden, so erhält man als Ergebnis die Steigung der ersten Ableitung, also den Funktionswert der zweiten Ableitung.

Wie Sie sicherlich bemerkt haben werden, befindet sich im ersten Menü noch ein Punkt mit der Bezeichnung '3-dimensional', der jedoch nicht wählbar ist. Möchten Sie auch Funktionen der linearen Algebra darstellen können, so ist Ihre eigene Initiative gefragt. Um den Menüpunkt aus seinem Dauerschlaf zu erwecken, müssen Sie lediglich im Unterprogramm 'setmenu' die Zeile 'Menu 2,2' löschen und an das Programm eine neue Prozedur anhängen, die diesen Aufgaben gerecht wird. Die meistgebrauchten Funktionen der linearen Algebra besitzen die Gestalten, die in Box 2 dargestellt sind.

Für Geraden sind also drei Funktionen der Gestalt x(i) = a(i) + r*b(i); (i = 1,2,3) zu definieren, wobei a(i) und b(i) eingegeben werden. Jeder Punkt der Gerade berechnet sich, indem man für r beliebige Werte aus der Menge der rationalen Zahlen einsetzt. Dabei ist zu beachten, daß man mit jeder Gleichung x(i) immer nur eine Koordinate des Punktes erhält, also daß x(1) nur die erste Koordinate des Punktes x ist.
Es ist durchaus sinnvoll, das zweidimensionale Koordinatensystem beizubehalten, und den Betrachtungswinkel immer um 90° zu ändern. So erspart man sich lästiges Umrechnen in die dritte (nicht vorhandene) Dimension.

(br)

Geraden:

$$\vec{x} = \vec{a} + r * \vec{b}$$

Ebenen:

$$\vec{x} = \vec{a} + r\vec{b} + s * \vec{c}$$

r,s Element R
a,b,c Element R∞
' → ' (sprich Vektor)

Box 2

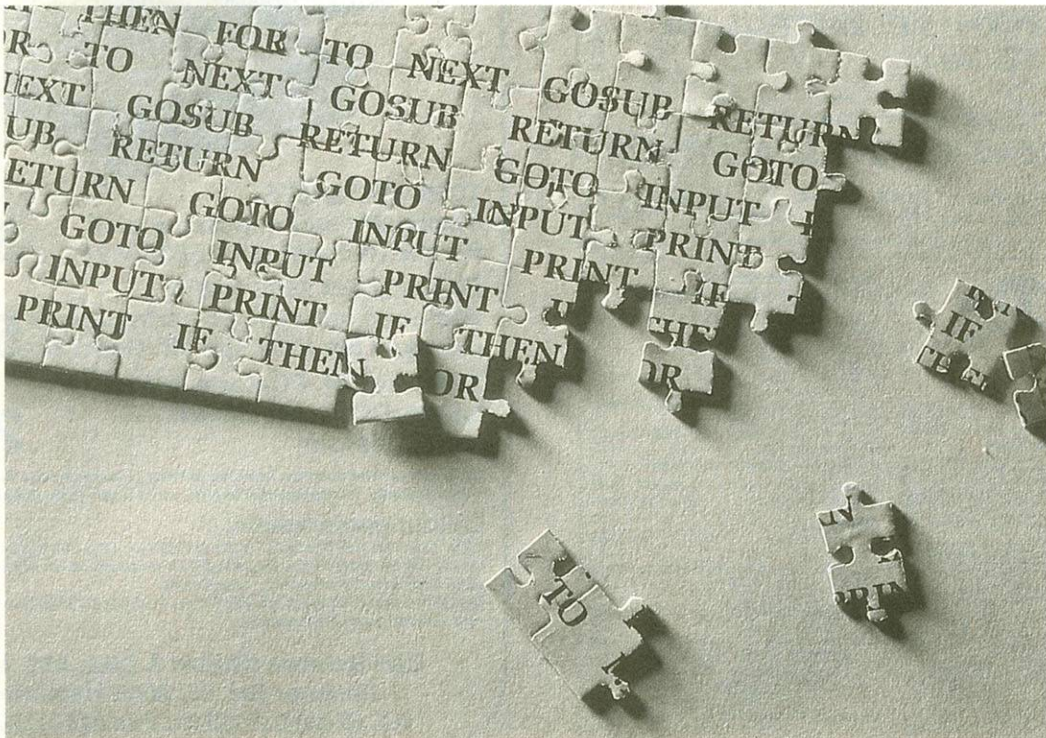
Listings

```
REM Funktionsplotter
RESERVE 100000
' ------ Definitionen -----------
DIM fa(30,3),bla%(3)
DEFFN selbstdef(x)=x^4-4*x^2+3
DEFFN x(x)=fa(1,1)*x
DEFFN x2(x)=fa(2,1)*x^2
DEFFN x3(x)=fa(3,1)*x^3
DEFFN x4(x)=fa(4,1)*x^4
DEFFN xn(x)=fa(5,1)*x^(fa(5,2))
DEFFN xd1(x)=fa(6,1)*1/x
DEFFN xd1n(x)=fa(7,1)*1/x^fa(7,2)
DEFFN sqrx(x)=fa(8,1)*SQR(ABS(x))
DEFFN sinx(x)=fa(9,1)*SIN(x)
DEFFN cosx(x)=fa(10,1)*COS(x)
DEFFN tanx(x)=fa(11,1)*TAN(x)
DEFFN ex(x)=fa(12,1)*EXP(x)
DEFFN v(x)=FN x(x)+FN x2(x)+FN x3(x)+FN x4(x)+FN xn(x)
+FN xd1(x)+FN xd1n(x)+FN sqrx(x)+FN sinx(x)+FN cosx(x)
+FN tanx(x)+FN ex(x)+fa(0,1)
m%=0
dimens%=0
xli%=-10
xri%=10
yui%=-10
yoi%=10
stufx%=31
stufy%=11
erst%=1
selbst%=1
inter%=0
abl%=0
ARRAYFILL bla%(),0
' -------Ende des Deklarationsteils
ON ERROR GOSUB fehler
windopen
DEFMOUSE 5
SETCOLOR 0,0,0,0
SETCOLOR 1,15,15,15
SETCOLOR 2,15,10,0
SETCOLOR 3,15,0,0
SETCOLOR 4,0,0,15
SETCOLOR 5,15,5,0
SETCOLOR 6,15,5,5
SETCOLOR 7,13,13,13
COLOR 2,0,1
setmenu
loeschen(0)
restart:
mx=20
my=10
mz=30
ON MENU GOSUB menuselect
REPEAT
  SLEEP
UNTIL m%=3
IF m%<>3 THEN
  GOTO restart
ENDIF
windclose
END
' ----------------------------------------------------------
PROCEDURE dimension
  dimens%=1-dimens%
  IF dimens%=1 THEN
    MENU 2,"  3-dimensional"
  ELSE
    MENU 2,"  2-dimensional"
  ENDIF
RETURN
'
PROCEDURE setmenu
  LOCAL i%
  DIM a$(40)
  REPEAT
    READ a$(i%)
    PRINT a$(i%)
    INC i%
  UNTIL a$(i%-1)="XXX"
  a$(i%-1)=""
  DATA FunktionsPlot  ,  Help,  3-dimensional,  Quit,
  DATA Layout  ,  Zeichnen ,  Ableiten ,  Intervall ,
 Löschen ,  Print ,
  DATA Funktionen  ,  Erstellen ,  Konst,  x,  x^2,  x
^3,  x^4,  x^n,  1/x,  1/(x^n),  SQR(),  sin(),  cos()
,  tan(),  e^x,  Vordef.,
  DATA  Tools  ,  Lineal ,  Interv ,  f() ,
  DATA XXX
  MENU a$()
  MENU 2,2
  MENU 31,64+16+8++2+1
  FOR i%=14 TO 27
    MENU i%,2
  NEXT i%
RETURN
'
PROCEDURE menuselect
  m%=MENU(0)
  IF m%<>3 THEN
    IF m%<13 THEN
      IF m%=9 THEN
        GOSUB loeschen(0)
      ELSE
        ON m% GOSUB help,dimension,a,a,a,zeichnen,abl,
intervall,loeschen,prt
      ENDIF
    ELSE
      m%=m%-12
      ON m% GOSUB erstell,konst,x,x2,x3,x4,xn,1x,1xn,s
qrx,sinx,cosx,tanx,ex,vordef,a,a,lineal,interv,fwert
      m%=m%+11
    ENDIF
  ENDIF
RETURN
'
PROCEDURE interv
  inter%=1-inter%
  IF inter%=1 THEN
    intwindop
```

```
    interprint
  ELSE
    intwindclos
  ENDIF
RETURN
'
PROCEDURE windopen
  OPENS 1,0,0,640,256,3,&H8002
  OPENW 1,0,0,640,256,&H300,&HC08,1
RETURN
'
PROCEDURE intwindop
  OPENW 3,550,220,88,30,0,&H802,1
  OPENW 1
RETURN
'
PROCEDURE intwindclos
  CLOSEW 3
RETURN
'
PROCEDURE interprint
  OPENW 3
  CLS
  PRINT "x[";xli%;",";xri%;"]"
  PRINT "y[";yui%;",";yoi%;"]";
  OPENW 1
RETURN
'
PROCEDURE mitwindopen
  OPENW 2,100,100,400,70,0,0,1
RETURN
'
PROCEDURE mitwindclose
  CLOSEW 2
RETURN
'
PROCEDURE windclose
  intwindclos
  CLOSEW 1
  CLOSES 1
RETURN
'
' ---------------------------------------------------
PROCEDURE zeichnen
  bla%(0)=1
  abl%=0
  x%=10
  z%=0
  FOR i=xli% TO xri% STEP 1/stufx%
    IF selbst%=0 THEN
      y=ykoord%-stufy%*FN v(i)
    ELSE
      y=ykoord%-stufy%*FN selbstdef(i)
    ENDIF
    IF -20<y AND y<244 AND z%<>0 THEN
      DRAW  TO x%,y
    ELSE
      IF 0<y AND y<244 THEN
        DRAW x%,y TO x%,y
        z%=1
      ELSE
        z%=0
      ENDIF
    ENDIF
    INC x%
  NEXT i
RETURN
'
PROCEDURE intervall
  OPENW 2,160,50,300,150,0,0,1
  OPENW 2
  REPEAT
    CLS
    PRINT "X-Achse:"
    LOCATE 32,2
    PRINT "[";xli%;"]"
    LOCATE 0,2
    INPUT "Linke  Intervallgrenze : ",x$
    IF x$<>"" THEN
      xli%=VAL(x$)
    ENDIF
    LOCATE 32,3
    PRINT "[";xri%;"]"
    LOCATE 0,3
    INPUT "Rechte Intervallgrenze : ",x$
    IF x$<>"" THEN
      xri%=VAL(x$)
    ENDIF
    PRINT
    PRINT "Y-Achse:"
    LOCATE 32,5
    PRINT "[";yui%;"]"
    LOCATE 0,5
    INPUT "Untere Intervallgrenze  : ",x$
    IF x$<>"" THEN
      yui%=VAL(x$)
    ENDIF
    LOCATE 32,6
    PRINT "[";yoi%;"]"
    LOCATE 0,6
    INPUT "Obere Intervallgrenze  : ",x$
    IF x$<>"" THEN
      yoi%=VAL(x$)
    ENDIF
  UNTIL xli%<xri% AND yui%<yoi%
  CLOSEW 2
  stufx%=INT(620/(ABS(xli%-xri%)))
  stufy%=INT(228/(ABS(yoi%-yui%)))
  loeschen(0)
RETURN
'
PROCEDURE loeschen(q%)
  OPENW 1
  CLEARW 1
  CLS
  koordsys
  IF q%=0 THEN
    ARRAYFILL bla%(),0
  ENDIF
RETURN
'
PROCEDURE prt
  SETCOLOR 0,15,15,15
  SETCOLOR 1,0,0,0
  SETCOLOR 2,0,0,0
  SETCOLOR 3,4,4,4
  SETCOLOR 4,0,0,0
  SETCOLOR 5,8,8,8
  loeschen(1)
  IF bla%(0) THEN
    zeichnen
  ENDIF
  FOR br%=1 TO 3
    abl%=bla%(br%)
    ableiten(abl%)
  NEXT br%
  HARDCOPY
  SETCOLOR 0,0,0,0
  SETCOLOR 1,15,15,15
  SETCOLOR 2,15,10,0
  SETCOLOR 3,15,0,0
  SETCOLOR 4,0,0,15
  SETCOLOR 5,15,5,0
  SETCOLOR 6,15,5,5
  COLOR 2,0,1
  loeschen(1)
  IF bla%(0) THEN
    zeichnen
  ENDIF
  FOR br%=1 TO 3
    abl%=bla%(br%)
    ableiten(abl%)
  NEXT br%
RETURN
'
PROCEDURE koordsys
  xkoord%=10-stufx%*xli%
  ykoord%=8+stufy%*yoi%
  DRAW xkoord%,0 TO xkoord%,256
  DRAW 0,ykoord% TO 640,ykoord%
  IF dimens%=1 THEN
    REM Bezeichnungen ändern
  ELSE
    IF ykoord%>0 THEN
      TEXT 630,ykoord%+10,"x"
    ENDIF
    IF xkoord%>0 THEN
      TEXT xkoord%+10,6,"y"
    ENDIF
  ENDIF
  IF ykoord%>244 THEN
    y%=246
  ELSE
    y%=ykoord%
  ENDIF
  FOR i%=0 TO ABS(xri%)+ABS(xli%)
    DRAW 10+i%*stufx%,y%-2 TO 10+i%*stufx%,y%+2
  NEXT i%
  IF xkoord%<0 THEN
    x%=1
  ELSE
    x%=xkoord%
  ENDIF
  FOR i%=0 TO ABS(yui%)+ABS(yoi%)
    DRAW x%-2,8+i%*stufy% TO x%+2,8+i%*stufy%
  NEXT i%
  IF inter%=1 THEN
    interprint
  ENDIF
RETURN
'
PROCEDURE erstell
  erst%=1-erst%
  IF erst%=0 THEN
    MENU 13,"  Beenden"
    FOR i%=14 TO 27
      MENU i%,64+16+2
    NEXT i%
    MENU 1,2
    MENU 3,2
    FOR i%=6 TO 10
      MENU i%,2
    NEXT i%
    mitwindopen
    ARRAYFILL fa(),0
    PRINT "Erstellen einer Funktion:"
    PRINT "f(x)= ";
    OPENW 1
    zeich%=1
  ELSE
    MENU 13,"  Erstellen"
    FOR i%=14 TO 27
      MENU i%,2
    NEXT i%
```

```
      MENU 1,82
      MENU 3,82
      FOR i%=6 TO 10
        MENU i%,82
      NEXT i%
      mitwindclose
    ENDIF
RETURN
'
PROCEDURE konst
  fa(0,1)=FN nn
  PRINT faktor;"+"
  OPENW 1
  zeich%=zeich%+LEN(STR$(faktor))+1
RETURN
'
PROCEDURE x
  fa(1,1)=FN nn
  PRINT faktor;"*x+"
  OPENW 1
  zeich%=zeich%+LEN(STR$(faktor))+3
RETURN
'
PROCEDURE x2
  fa(2,1)=FN nn
  PRINT faktor;"*x^2+"
  OPENW 1
  zeich%=zeich%+LEN(STR$(faktor))+5
RETURN
'
PROCEDURE x3
  fa(3,1)=FN nn
  PRINT faktor;"*x^3+"
  OPENW 1
  zeich%=zeich%+LEN(STR$(faktor))+5
RETURN
'
PROCEDURE x4
  fa(4,1)=FN nn
  PRINT faktor;"*x^4+"
  OPENW 1
  zeich%=zeich%+LEN(STR$(faktor))+5
RETURN
'
PROCEDURE xn
  potn
  fa(5,1)=faktor
  fa(5,2)=faktor2
  PRINT fa(5,1);"*x^";fa(5,2);"+"
  OPENW 1
  zeich%=zeich%+LEN(STR$(fa(5,1)))+LEN(STR$(fa(5,2)))+
4
RETURN
'
PROCEDURE potn
  selbst%=0
  OPENW 2
  LOCATE 0,4
  INPUT "Faktor: ",faktor$
  INPUT "Exponent: ",faktor2$
  faktor=VAL(faktor$)
  faktor2=VAL(faktor2$)
  LOCATE 0,4
  PRINT "                                        "
  PRINT "                                        "
  vorzeich
  LOCATE 7+zeich%,2
RETURN
'
FUNCTION nn
  selbst%=0
  OPENW 2
  LOCATE 0,4
  INPUT "Faktor: ",faktor$
  faktor=VAL(faktor$)
  LOCATE 0,4
  PRINT "                                        "
  vorzeich
  LOCATE 7+zeich%,2
  RETURN faktor
ENDFUNC
'
PROCEDURE vorzeich
  IF faktor+ABS(faktor)=0 THEN
    zeich%=zeich%-1
  ENDIF
RETURN
'
PROCEDURE 1x
  fa(6,1)=FN nn
  vorzeich
  PRINT faktor;"*1/x+"
  OPENW 1
  zeich%=zeich%+LEN(STR$(faktor))+5
RETURN
'
PROCEDURE 1xn
  potn
  fa(7,1)=faktor
  fa(7,2)=faktor2
  PRINT fa(7,1);"*1/x^";fa(7,2);"+"
  OPENW 1
  zeich%=zeich%+LEN(STR$(fa(7,1)))+LEN(STR$(fa(7,2)))+
6
RETURN
'
PROCEDURE sqrx
  fa(8,1)=FN nn
  PRINT faktor;"*SQR(x)+"
  OPENW 1
  zeich%=zeich%+LEN(STR$(faktor))+8
RETURN
'
PROCEDURE sinx
  fa(9,1)=FN nn
  PRINT faktor;"*SIN(x)+"
  OPENW 1
  zeich%=zeich%+LEN(STR$(faktor))+8
RETURN
'
PROCEDURE cosx
  fa(10,1)=FN nn
  PRINT faktor;"*COS(x)+"
  OPENW 1
  zeich%=zeich%+LEN(STR$(faktor))+8
RETURN
'
PROCEDURE tanx
  fa(11,1)=FN nn
  PRINT faktor;"*TAN(x)+"
  OPENW 1
  zeich%=zeich%+LEN(STR$(faktor))+8
RETURN
'
PROCEDURE ex
  fa(12,1)=FN nn
  PRINT faktor;"*e^x+"
  OPENW 1
  zeich%=zeich%+LEN(STR$(faktor))+8
RETURN
'
PROCEDURE vordef
  selbst%=1
  erstell
RETURN
'
PROCEDURE lineal
  LOCAL x%,y%
  GET 0,0,320,244,b$
  GET 321,0,639,244,c$
  COLOR 4
  OPENW 1
  REPEAT
    x%=MOUSEX
    y%=MOUSEY
    DRAW 0,y% TO 640,y%
    DRAW x%,0 TO x%,244
    LOCATE 0,0
    PRINT "x: ";
    PRINT USING "+##.##",(x%-xkoord%)/stufx%
    PRINT "y: ";
    PRINT USING "+##.##",-(y%-ykoord%)/stufy%
    DO
      maus%=MOUSEK
    LOOP WHILE x%=MOUSEX AND y%=MOUSEY AND maus%=0
    PUT 0,0,b$
    PUT 321,0,c$
  UNTIL maus%=1
  COLOR 2,0
RETURN
'
PROCEDURE abl
  mitwindopen
  LOCATE 0,2
  INPUT "Welche Ableitung (1-3): ",abl%
  mitwindclose
  ableiten(abl%)
RETURN
'
PROCEDURE ableiten(abl%)
  IF abl%=1 THEN
    bla%(1)=1
    COLOR 5
    x%=10
    z%=0
    FOR i=xli% TO xri% STEP 1/stufx%
      IF selbst%=0 THEN
        y=ykoord%-stufy%*FN abl1(i)
      ELSE
        y=ykoord%-stufy%*FN selbst1(i)
      ENDIF
      IF -20<y AND y<244 AND z%<>0 THEN
        DRAW  TO x%,y
      ELSE
        IF 0<y AND y<244 THEN
          DRAW x%,y TO x%,y
          z%=1
        ELSE
          z%=0
        ENDIF
      ENDIF
      INC x%
    NEXT i
  ELSE
    IF abl%=2 THEN
      bla%(2)=2
      COLOR 3
      x%=10
      z%=0
      FOR i=xli% TO xri% STEP 1/stufx%
        IF selbst%=0 THEN
          y=ykoord%-stufy%*FN abl2(i)
```

```
          ELSE
            y=ykoord%-stufy%*FN selbst2(i)
          ENDIF
          IF -20<y AND y<244 AND z%<>0 THEN
            DRAW  TO x%,y
          ELSE
            IF 0<y AND y<244 THEN
              DRAW x%,y TO x%,y
              z%=1
            ELSE
              z%=0
            ENDIF
          ENDIF
          INC x%
        NEXT i
      ELSE
        IF abl%=3 THEN
          bla%(3)=3
          COLOR 7
          x%=10
          z%=0
          FOR i=xli% TO xri% STEP 1/stufx%
            IF selbst%=0 THEN
              y=ykoord%-stufy%*FN abl3(i)
            ELSE
              y=ykoord%-stufy%*FN selbst3(i)
            ENDIF
            IF -20<y AND y<244 AND z%<>0 THEN
              DRAW  TO x%,y
            ELSE
              IF 0<y AND y<244 THEN
                DRAW x%,y TO x%,y
                z%=1
              ELSE
                z%=0
              ENDIF
            ENDIF
            INC x%
          NEXT i
        ENDIF
      ENDIF
    ENDIF
    COLOR 2
RETURN
'
FUNCTION abl1(i)
  y1=FN v(i)
  y2=FN v(i+1/stufx%)
  y=(y2-y1)*stufx%
  RETURN y
ENDFUNC
'
FUNCTION selbst1(i)
  y1=FN selbstdef(i)
  y2=FN selbstdef(i+1/stufx%)
  y=(y2-y1)*stufx%
  RETURN y
ENDFUNC
'
FUNCTION abl2(i)
  y1=FN v(i)
  y2=FN v(i+1/stufx%)
  y3=FN v(i+2/stufx%)
  y4=(y2-y1)*stufx%
  y5=(y3-y2)*stufx%
  y=(y5-y4)*stufx%
  RETURN y
ENDFUNC
'
FUNCTION selbst2(i)
  y1=FN selbstdef(i)
  y2=FN selbstdef(i+1/stufx%)
  y3=FN selbstdef(i+2/stufx%)
  y4=(y2-y1)*stufx%
  y5=(y3-y2)*stufx%
  y=(y5-y4)*stufx%
  RETURN y
ENDFUNC
'
FUNCTION abl3(i)
  y1=FN v(i)
  y2=FN v(i+1/stufx%)
  y3=FN v(i+2/stufx%)
  y4=FN v(i+3/stufx%)
  y5=(y2-y1)*stufx%
  y6=(y3-y2)*stufx%
  y7=(y4-y3)*stufx%
  y8=(y6-y5)*stufx%
  y9=(y7-y6)*stufx%
  y=(y9-y8)*stufx%
  RETURN y
ENDFUNC
'
FUNCTION selbst3(i)
  y1=FN selbstdef(i)
  y2=FN selbstdef(i+1/stufx%)
  y3=FN selbstdef(i+2/stufx%)
  y4=FN selbstdef(i+3/stufx%)
  y5=(y2-y1)*stufx%
  y6=(y3-y2)*stufx%
  y7=(y4-y3)*stufx%
  y8=(y6-y5)*stufx%
  y9=(y7-y6)*stufx%
  y=(y9-y8)*stufx%
  RETURN y
ENDFUNC
'
PROCEDURE fwert
  mitwindopen
  LOCATE 0,2
  PRINT "(0) f(x),  (1) f'(x),  (2) f''(x),  (3) f'''(x)"
  REPEAT
    INPUT "Funktion: ",f$
    f=VAL(f$)
  UNTIL f=0 OR f=1 OR f=2 OR f=3
  mitwindclose
  LOCAL x%,y%
  COLOR 4
  REPEAT
    x%=MOUSEX
    y%=MOUSEY
    LOCATE 0,0
    x=(x%-xkoord%)/stufx%
    IF f=0 THEN
      PRINT "f(";
      PRINT USING "+##.##",x;
      PRINT ")=";
      IF selbst%=0 THEN
        PRINT USING "+###.##",ROUND(FN v(x),2)
      ELSE
        PRINT USING "+###.##",ROUND(FN selbstdef(x),2)
      ENDIF
    ELSE
      IF f=1 THEN
        PRINT "f'(";
        PRINT USING "+##.##",x;
        PRINT ")=";
        IF selbst%=0 THEN
          PRINT USING "+###.##",ROUND(FN abl1(x),2)
        ELSE
          PRINT USING "+###.##",ROUND(FN selbst1(x),2)
        ENDIF
      ELSE
        IF f=2 THEN
          PRINT "f''(";
          PRINT USING "+##.##",x;
          PRINT ")=";
          IF selbst%=0 THEN
            PRINT USING "+###.##",ROUND(FN abl2(x),2)
          ELSE
            PRINT USING "+###.##",ROUND(FN selbst2(x),2)
          ENDIF
        ELSE
          IF f=3 THEN
            PRINT "f'''(";
            PRINT USING "+##.##",x;
            PRINT ")=";
            IF selbst%=0 THEN
              PRINT USING "+###.##",ROUND(FN abl3(x),2)
            ELSE
              PRINT USING "+###.##",ROUND(FN selbst3(x),2)
            ENDIF
          ENDIF
        ENDIF
      ENDIF
    ENDIF
    DO
      maus%=MOUSEK
    LOOP WHILE x%=MOUSEX AND y%=MOUSEY AND maus%=0
  UNTIL maus%=1
  LOCATE 0,0
  PRINT "                          "
  COLOR 2,0
RETURN
'
PROCEDURE fehler
  ON ERROR GOSUB fehler
  RESUME NEXT
RETURN
'
PROCEDURE help
  OPENW 2,155,50,330,170,0,0,1
  OPENW 2
  PRINT "           Funktionsplotter"
  PRINT "           ----------------"
  PRINT
  PRINT "Help: Diese Übersicht"
  PRINT "Quit: Programmende"
  PRINT
  PRINT "Zeichnen : Funktion zeichnen"
  PRINT "Ableiten : 1. bis 3. Abl. zeichnen"
  PRINT "Intervall: Zeichenintervall definieren"
  PRINT "Print    : Bildschirm drucken"
  PRINT
  PRINT "Erstellen: Funktion definieren"
  PRINT "Beenden  : Definition abschließen"
  PRINT
  PRINT "Lineal: x- und y- Koordinaten der Maus"
  PRINT "Interv: Anzeige d. Intervalls  An/Aus"
  PRINT "f()   : Funktionswerte anzeigen"
  PRINT
  PRINT "                              <Taste>";
  REPEAT
    a$=INKEY$
  UNTIL a$<>""
  CLOSEW 2
  OPENW 1
RETURN
```



AMIGA-BASIC

Programmieren mit Erfolg

Punkt, Punkt, Komma, Strich...

... nicht ein Mondgesicht, sondern eine kleine Grafik könnte so relativ einfach und unkompliziert entstehen. Da wir ja bereits ein wenig Struktur in unseren Programmwust hineingebracht haben, stürzen wir uns heute auf das Gebiet des Amiga schlechthin: die Grafik. Wie lassen sich unter BASIC kunterbunte Muster oder hübsche Bilder erzeugen? Öffnen wir unsere BASIC-Kiste und sehen nach, mit welchen Befehlen sich derlei Bildchen oder ähnliches realisieren lassen.

Kursfahrplan



Beim Stichwort Grafik kommt so mancher Computerbesitzer ins Schwärmen, ist sie doch bei jedem Programm erst das Tüpfelchen auf dem i. Selbst eingefleischte PC-Freaks müssen dabei neidlos anerkennen, daß in puncto Grafik der Amiga fast allen privaten PCs einiges voraus hat (wenn es sich nicht gerade um einen speziellen Grafik-PC handelt). Grafiken anzuschauen ist dabei gar nicht schwer, Grafiken zu programmieren endet aber meist im untersten Bereich der Computersprachen, nämlich bei Assembler. Oder man macht es sich einfacher und zeichnet sich sein Bild mit einem bekannten Zeichenprogramm wie Deluxe Paint (ein kleiner Hinweis auf unseren Deluxe-Paint-Kurs sei gestattet) und liest es hinterher mit einem der bekannten IFF-Loader ein. So weit, so gut!

AmigaBASIC ist vielleicht nicht gerade das schnellste BASIC, jedoch bietet es eine Vielzahl an Befehlen, die für Grafik und Animation genutzt werden können. Sogar die Nutzung des Blitters ist in AmigaBASIC teilweise möglich, wie wir später noch sehen werden.

Für unsere Grafik benötigen wir ein Fenster, oder zwei, oder drei, oder...

Rufen wir uns noch einmal ins Gedächtnis zurück, was wir über Screens und Windows unter AmigaBASIC wissen:

– *Screens können eine beliebige Größe annehmen, die jedoch nur bis zur Maximalgröße reicht (Vollbild mit x1 = 0,y1 = 0,x2 = 640,y2 = 2 56, wenn wir auf den Interlace-Modus verzichten!), gleiches gilt für Windows.*

– *Screens benötigen eine bestimmte 'Tiefe', die für die Anzahl der verwendeten Farben notwendig ist (in BASIC bis zu 32 Farben gleichzeitig).*

– *Bis zu vier Screens lassen sich mit unterschiedlicher Auflösung realisieren.*

– *Programme können Texte und Grafiken auf alle erzeugten Windows umleiten, Ausgaben können daher auch unsichtbar vorgenommen werden.*

– *Windows und Screens sind sehr speicherintensiv, man sollte seinen Programmen also vorher etwas mehr Platz reservieren.*

Um die Wirkungsweise von Windows noch einmal zu demonstrieren, sollten Sie sich unser erstes Listing einmal genauer ansehen. Hier werden neben dem Hauptwindow in voller Größe noch zwei kleinere Windows erzeugt, in denen jeweils nacheinander ein Zähler abläuft. In der FOR-NEXT-Schleife bildet dabei die Variable 'w' das aktuelle WINDOW.

Das kleine 'Fensterauf-Fensterzu'-Programm soll uns noch einmal als Gedächtnisstütze dienen, um uns den WINDOW-Befehl wieder vor Augen zu führen.

Doch nun zur Grafik. Grafische Darstellungen auf dem Computer sind naturgemäß anders als die Zeichnung auf einem Blatt Papier. Wenn wir uns einen Zeichenblock und einen Bleistift kaufen, brauchen wir uns nur an einen freien Tisch zu setzen und den kleinen Picasso zu imitieren. Nun kommt es nur noch darauf an, ob sich Ihre Bilder so gut verkaufen wie die von Picasso – oder nur so aussehen.

Beim Zeichnen auf dem Monitor unseres Amigas nutzt der Bleistift recht wenig, es sei denn, Sie gebrauchen den Monitor als 'Notepad' (denken Sie daran, Radiergummies ziehen Streifen auf Glas). Bei einer Computergrafik sind 'Pixel' das bestimmende Element. 'Pixel' ist wieder eines von diesen Kunstwörtern, die sich aus mehreren Begriffen zusammensetzen: 'PIcture (X) ELement', wobei das 'X' für das 'geknautschte Piktschör' steht. Ein Pixel ist die kleinste grafische Einheit eines Bildschirms, und doch haben sämtliche Grafikbefehle etwas mit diesem Pixel zu tun.

In AmigaBASIC läßt sich dieses Pixel direkt setzen, die Befehle dazu lauten PSET und PRESET. PSET und PRESET brauchen die Koordinaten, an denen ein Punkt gesetzt werden soll, sowie eine Angabe zur aktuellen Farbe des Pixels. Die Syntax der beiden Befehle sieht damit so aus:

PSET (xKoordinate,yKoordinate),Farbe

oder

PRESET (xKoordinate,yKoordinate),Farbe

Zum PSET- oder PRESET-Befehl kann noch STEP angegeben werden, dann beziehen sich die x- und y-Koordinaten nicht absolut auf die X-/Y-Achse, sondern relativ. Falls dies zu verwirrend sein sollte, habe ich hier ein kurzes Beispiel:

Wenn wir vier Eckpunkte auf dem Bildschirm setzen wollen, können wir dies auf zwei Arten hinbekommen. Art Nummer 1 nennt sich ABSOLUT und bezieht sich immer auf einen vordefinierten 0-Punkt (der sinnigerweise an den Koordinaten x = 0,y = 0 liegt):

```
PSET (0,0),3
PSET (100,0),3
PSET (100,100),3
PSET (0,100),3
```

Nach dem Start des Programms bekommen wir unsere Punkte tatsächlich an die Eckpunkte eines imaginären Rechtecks gesetzt:

```
° (0,0) (100,0)       °
° (0,100) (100,100) °
```

Das gleiche können wir aber auch RELATIV machen:

```
PSET (0,0),2
PSET STEP (100,0),2
PSET STEP (0,100),2
PSET STEP (-100,0),2
```

```
° (0,0) (100,0)        °
° (-100,0) (0,100)   °
```

Der Unterschied wird erst in den letzten beiden Zeilen deutlich, auch wenn Zeile 2 schon zum relativen Bezugspunkt berechnet wurde. Hierbei wird nämlich der zuletzt gezeichnete Punkt zum aktuellen Nullpunkt. Gehen wir das zweite Listing doch mal durch:

PSET (0,0) setzt den ersten Punkt ganz normal an die Koordinaten x = 0, y = 0. Der zweite Punkt wird nun durch STEP auf der x-Achse um 100 Pixel zur positiven Seite hin (nach rechts, denn unser Bildschirm hat ja bis zu insgesamt 640 Punkte) versetzt, während sich auf der y-Achse nichts tut. Beim dritten PSET-Befehl ändert sich diesmal nichts auf der x-Achse, dafür wird der Punkt um 100 Pixel nach unten auf der y-Achse gesetzt. Unser aktuelles x/y-Verhältnis lautet nun so: x = 100/y = 100. Um nun den vierten Punkt zu setzen, der 100 Pixel unter dem Anfangspunkt liegt, müssen wir 100 Pixel auf der x-Achse zurückgehen, daher die Angabe '-100,0'.

Durch die unterschiedlichen Farben (2 und 3) sieht man, daß bei beiden Methoden das gleiche Ergebnis zu Tage kommt.

Wozu jedoch die Unterscheidung von absoluten oder relativen Koordinaten? Nun, wenn man in BASIC ein gleichmäßiges Vieleck (Sechseck, Achteck...) zeichnen will, so lassen sich die Eckpunkte sehr schnell durch relative Koordinaten bestimmen.

Dazu ein Beispiel (indem wir ein bißchen auf den Befehlsvorrat vorgreifen, Erklärungen folgen später im Text):

```
WINDOW 1:CLS
AREA (200,120)
AREA STEP (60,0)
AREA STEP (25,-25)
AREA STEP (-25,-25)
AREA STEP (-60,0)
AREA STEP (-25,25)
AREAFILL 1
WHILE INKEY$="":WEND
AREA (200,120)
AREA (260,120)
AREA (285,95)
AREA (260,70)
AREA (200,70)
AREA (175,95)
AREAFILL 0
```

Unser kleines Programm zaubert zwei Sechsecke übereinander auf den Bildschirm, das zweite erst nach Tastendruck und in anderer Farbe.

Während das erste Sechseck durch relative Koordinaten bestimmt wird, errechnet sich Nummer zwei absolut. Wenn wir auf die Koordinaten von Sechseck eins schauen, dann sehen wir, daß sie abhängig vom jeweils letzten Punkt errechnet wurden, während die Punkte von Sechseck zwei immer als Bezugspunkt x = 0,y = 0 haben. Welche Art von Koordinaten Sie nehmen, sei Ihnen überlassen, aber

versuchen Sie doch selbst einmal, ein auf einer Ecke stehendes Sechseck oder vielmehr die Eckpunkte davon mit relativen Koordinaten zu programmieren. Viel Glück!
Ein wichtiges Detail bei PSET und PRESET ist noch zu erwähnen: der Unterschied. Im ersten Augenblick ist er nicht sofort augenscheinlich, läßt man jedoch die Angabe der Farbe bei beiden Befehlen weg, so wird bei PSET automatisch die Zeichenfarbe, bei PRESET die Hintergrundfarbe angenommen.
Ein kleines Beispiel zum Einsatz von PSET finden Sie in Listing 2.

Punkt, Punkt, Komma haben wir! Fehlt noch ...

...genau, Striche. Diese in AmigaBASIC zu zeichnen, ist denkbar einfach, Beispiele dazu haben wir schon in den vergangenen Kursen aufgezeigt. Der Befehl dazu heißt LINE. Er braucht zum korrekten Ablauf noch Angaben über Start- und Endpunkt. Diese Angaben sind ähnlich wie bei PSET/PRESET-Befehl:

LINE (xAnfang,yAnfang)-(xEnde,yEnde),Farbe

LINE (1,1)-(100,100),3 zeichnet also eine Linie von der x/y-Koordinate 1,1 zur x/y-Koordinate 100,100. So weit, so gut. Zeichnen wir doch einmal ein Rechteck. Also fix eingegeben: LINE dies nach da, LINE da nach dort, LINE dort...
Nein, so nicht! Beim LINE-Befehl haben die Entwickler des BASIC-Interpreters schon an die armen Programmierer gedacht und ihn (den Befehl) mit ein paar Extras ausgestattet. Hängt man zum Beispiel an den Farbwert noch ein 'B' an, so wird anstelle eines Striches eine Box gezeichnet, fügt man darüber hinaus noch ein 'F' an, gibt's dazu noch eine gefüllte Box.
Probieren wir es mal mit einem kleinen Programm:

```
Start:
  WINDOW 1:CLS:GOSUB Werte
  WHILE INKEY$=""
    LINE (x1,y1)-(x2,y2),fb,b
    IF x1<320 THEN
      x1=x1+4
      x2=x2-4
      y1=y1+2
      y2=y2-2
      fb=fb+1:IF fb>3 THEN
      b=0
```

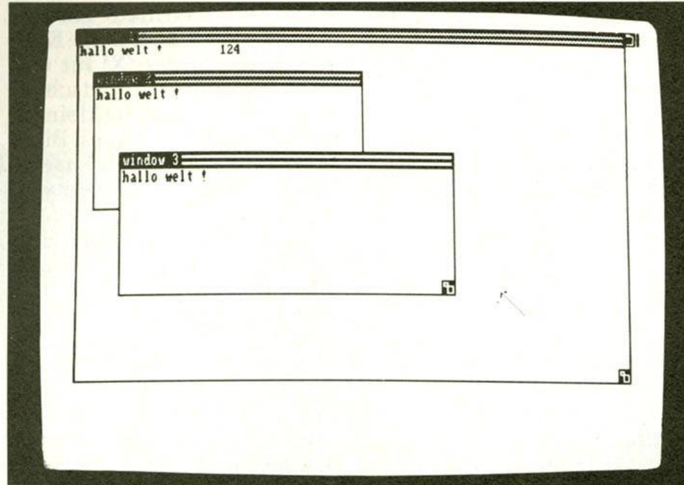


Bild 1: Listing 1 demonstriert eine der wichtigsten Funktionen, das Errichten von neuen Windows

```
      ELSE
    GOSUB Werte
   END IF
  WEND
Werte:
x1=1:x2=640:y1=1:y2=200:fb=0
RETURN
```

Ein buntes Rahmengemisch ist auf dem Bildschirm zu sehen, wenn Sie das kleine Programm starten.
Ändern Sie jetzt den Parameter 'b' in der LINE-Zeile doch mal in 'bf' um und starten Sie das Programm erneut! Nun, können Sie noch gucken? Es flackert, daß es eine Freude ist, und zwar deshalb, weil Sie aus Rahmen nun farbige Boxen gemacht haben.
Wenn Sie noch ein wenig mit dem LINE-Befehl experimentieren wollen, schauen Sie sich doch mal Listing 3 und 4 an.

Blitter in BASIC – geht denn das? Hmm – Jein!

Eine der hervorragendsten Eigenschaften des Amiga sind seine Customchips. Der Grafikchip 'Agnus' enthält zwei Elemente, die in der Grafikprogrammierung eine sehr große Rolle spielen: der Copper und der Blitter. Was beide so leisten können, soll hier nicht unser Thema sein, dies läßt sich wesentlich besser in Assembler programmieren. Wir wollen sehen, ob BASIC die Eigenschaften des Grafikchips nutzen kann. Nun, die Antwort ist nicht gerade einfach. Wie schon gesagt, ist die Programmierung des Blitters nur mit entsprechender Kenntnis der Hardware möglich. Unter BASIC geht dies unter anderem auch über die PEEK- und POKE-Befehle, die direkt auf Speicheradressen zugreifen können. Jedoch gibt es eine Möglichkeit, Bildausschnitte schnell zu bearbeiten und dies mit Hilfe des Blitters.
Schauen wir uns doch mal Listing 5 an. Nach dem Start werden eine Menge lustiger Gesichter auf den Bildschirm gezaubert, nur leider recht langsam. Listing 6 dagegen macht genau das gleiche, allerdings mit zwei neuen Befehlen namens GET und PUT.
GET und PUT machen genau das, was sie beschreiben, GET übernimmt einen Teil des Grafikbildschirms und PUT setzt ihn an anderer Stelle wieder ein. Die Syntax des Befehls lautet:

GET(x1,y1)-(x2,y2),Feldvariable[Index1,Index2..]

(x1,y1) und (x2,y2) grenzen den Bereich ein, aus dem die Bitinformationen eingelesen werden. Der Bereich liegt dabei genauso wie beim LINE B oder BF-Befehl, also linke obere bis rechte untere Ecke.
Die Feldvariable muß angegeben werden, um die Bits, die vom GET-Befehl übernommen werden, unterzubringen. Diese Feldvariable muß natürlich auch den entsprechenden Platz bekommen, also durch den BASIC-Befehl DIM angelegt werden.
Wie groß muß aber nun die Feldvariable (nennen wir sie ab sofort der Einfachheit halber 'a%' (Integer-Variable)) werden? Immerhin wissen wir zwar genau, wie groß der Bildausschnitt werden soll, denn x1,y1,x2 und y2 sind von uns definierte Werte, jedoch geben sie keinerlei Auskunft darüber, wie groß a% sein muß. Im AmigaBASIC-Handbuch steht dazu eine Formel:

*6 + (y2-y1 + 1)*2*INT((x2-x1 + 16)/16)*T*

Wenn wir diese Formel in unser Programm einbauen, und zwar direkt an den Anfang, so können wir uns die Rechnerei für die DIM-Anweisung jedesmal sparen, wir müssen nur wissen, welchen Bereich wir in die GET-Anweisung einlesen wollen (x,y).
INDEX steht für weitere Feldvariablen, mit GET ist es also möglich, gleichzeitig mehrere Bildbereiche einzulesen.
PUT sorgt nun dafür, daß der in GET eingelesene Bit-Bereich wieder auf den Bildschirm gebracht wird. Die Syntax des Befehls sieht so aus:

PUT [STEP] (x,y), Feldvariable (Index,Index...Index,MODUS

(x,y) sind die Koordinaten, an denen unser Grafikausschnitt erscheinen soll, durch STEP lassen sich diese wieder relativ berechnen, die Feldvariable enthält die von GET angelegten Bitinformationen, Index steht wieder für weitere Feldvariablen (es können also wieder mehrere Grafikelemente zugleich plaziert werden) und Modus... Ja, Modus ist eigentlich das interessanteste am PUT-Befehl, hier können wir nämlich festlegen, wie unsere Grafik auf dem Hintergrund dargestellt werden soll.
In Modus lassen sich nämlich logische Verknüpfungen einsetzen, genauer gesagt AND, OR und XOR. Überlegen wir doch einmal, was wir über logische Verknüpfungen wissen (falls hierbei ein leeres Loch als Ergebnis herauskommen sollte, wird es Zeit, sich ein bißchen näher mit logischen Verknüpfungen und Bool'scher Algebra zu beschäftigen – schließlich basiert unser Computer ja darauf!!):
Logische Verknüpfungen werden dann benötigt, wenn mit zwei (oder mehreren) Binärzahlen gerechnet werden soll. Durch logische Verknüpfungen lassen sich aber auch die einzelnen Bits beeinflussen. Welche Art von logischen Verknüpfungen gibt es nun? Antwort: NOT, AND, OR, XOR. NOT ist die Invertierung eines Bits. War dieses vorher '0', wird es nach einer NOT-Verknüpfung '1', war es

'1', wird es danach '0'. Bei der UND-(AND-) Verknüpfung sieht die Sache schon schwieriger aus: Ein nicht gesetztes Bit, das mit einem nicht gesetzten Bit verknüpft wird, bleibt nicht gesetzt; ein gesetztes Bit, das mit einem nicht gesetzten Bit verknüpft wird, wird zu einem nicht gesetzten Bit; nur zwei gesetzte Bits miteinander verknüpft ergeben ein gesetztes Bit. Alles klar?

Wer jetzt erst mal aus dem Fenster schaut und tief Luft holt, dem sei verziehen. Das Ganze ist dabei aber gar nicht so schwer, schauen wir uns eine Verknüpfungstabelle für UND mit zwei Bits an:

Bit-1	*Bit-2*	*Erg.-Bit*
0	0	0
1	0	0
0	1	0
1	1	1

Als einfache Formel kann man sich folgendes merken: Das Ergebnisbit ist nur dann '1' ('An', 'High' oder wie immer man es nennen mag), wenn beide Verknüpfungsbits '1' sind. (Warum das so ist, ist nicht so einfach zu verstehen, wer es trotzdem wissen will, sollte sich Fachbücher über Digitalelektronik durchlesen, für unsere Zwecke reicht es zu wissen, daß es so ist!).

Kommen wir zum ODER. Hier ist das Ergebnis dann '1', wenn *mindestens ein Eingangsbit '1' ist.* Die Tabelle sieht dann so aus:

Bit-1	*Bit-2*	*Erg.-Bit*
0	0	0
1	0	1
0	1	1
1	1	1

Das sieht schon anders aus, nicht wahr? Ganz komisch wird es, wenn wir uns XOR ansehen (XOR steht für 'Exklusiv-ODER'!) dann sieht die Tabelle nämlich so aus:

Bit-1	*Bit-2*	*Erg.-Bit*
0	0	0
1	0	1
0	1	1
1	1	0

Hier ist das Ergebnisbit dann '1', wenn beide Eingangsbits ungleich sind.

Außer diesen Verknüpfungen kennt AmigaBASIC noch das Inhibit, die Äquivalenz und ähnliche Bitereien, da PUT jedoch nur die Genannten akzeptiert, wollen wir die anderen erst mal vergessen.

Was hat das jetzt alles mit dem PUT-Befehl zu tun? Nun, der Parameter MODUS erlaubt die Anwendung einer der vorgenannten Bitverknüpfungen zwischen Vordergrundgrafik (das GET-Feld) und Hintergrund (-Grafik). Am besten experimentieren Sie mal mit Listing 6 und setzen die logischen Verknüpfungen einfach ein.

Punkt, Komma und Strich ist da, aber Gesichter sind meistens rund, oder?

Richtig, Gesichter sind meistens rund, und damit sind wir beim nächsten Grafikbefehl:

CIRCLE STEP (x,y),radius,F arbe,Start,Ende,Bild

Circle zaubert (was auch sonst) Kreise auf den Bildschirm und zwar mit dem Mittelpunkt (x,y) und dem Radius 'radius'. Die x,y-Werte können durch STEP wieder relativ bezogen sein. (x,y) und 'radius' müssen angeben werden. Der Farbwert der 'Farbe' entspricht allen anderen Farbanweisungen.

'Start' und 'Ende' bezeichnen Anfang und Ende eines zu zeichnenden Kreises, man kann also mit dem Befehl auch Dreiviertel- oder Halbkreise zeichnen. Die beiden Werte 'Start' und 'Ende' sind allerdings nicht ganz einfach zu benutzen, hier sollte man noch einmal im Mathematikbuch nachschauen, was über Kreisbögen und deren Berechnung zu lesen ist (schon mal was von PI gehört?).

Im Ernst, 'Start' und 'Ende' sind Bogenmaßwerte, das heißt, sie sind Anfangs- und Endpunkt eines Kreises. Diese Punkte lassen sich normalerweise in Grad berechnen, jedoch waren die Entwickler des AmigaBASIC so freundlich, uns mit Bogenmaß zu beglücken. Also müssen wir wohl oder übel in den sauren Apfel beißen und umrechnen: Die Einheit des Bogenmaßes ist PI (3.141527... Sie wissen schon...) Ein PI (Bm) entspricht 180°. Ein Kreis hat 360°, also 2*PI, ein Viertelkreis 90°, also PI/2, ein Dreiviertelkreis 270° und damit 1.5*PI. Nun, programmieren wir doch einmal einen Kreis aus vier Viertelkreisen:

Das erste Segment fängt bei 0° (oder 360°, je nach Auffassung) an und endet bei 90°, das zweite fängt bei 90° an und endet bei 180°, wo dann Segment drei anfängt, welches bis 270° geht, danach schließt sich Segment vier von 270° bis 360° (oder 0°, wir hatten das schon...) an. Programmieren wir also:

```
WINDOW 1:CLS
REM Vier Kreissegmente(0-3
)=ein Vollkreis
PI=3.14:Kreisviertel=.5:'2
*PI sind 360°
Segment=0:'Mit diesem woll
en wir beginnen!
Farbe=1
WHILE Segment < 2
  Start=Segment*PI
  Ende=(Segment+Kreisvier
tel)*PI
  CIRCLE(320,100),90,Farb
e,Start,Ende
  Segment=Segment+Kreisvi
ertel
  Farbe=Farbe+1:IF Farbe>
3 THEN Farbe=2
WEND
```

Es lebe die Durchschaubarkeit! Gehen wir besser die Zeilen durch. In Zeile 1 machen wir das Ausgabe-Window frei, in Zeile zwei legen wir unsere Grundwerte fest:

– bekannt, zwei Stellen nach dem Komma reichen, Kreisviertel – enthält die Konstante, mit der PI multipliziert werden muß. Da wir Viertelkreise brauchen, ist Kreisviertel = 2 / 4, also 0.5 (2*PI sind 360°, also Kreis). Wenn wir Achtelkreise wollen, rechnen wir die Variable Kreisachtel aus: 2 / 8 = 0.25, bei Sechzehntelkreisen 2 / 16 = 0.125. Da wir mit dem ersten Kreisteil beginnen wollen, dieser jedoch bei 0° beginnt, bekommen die Kreissegmente die Nummern 0 bis 3 (also doch vier). Dieses erste Segment geht bis 90°, also muß das erste Kreisviertel dazugerechnet werden, das zweite geht bis 180° und so weiter, bis der Kreis geschlossen ist. – Die Farbe macht die Segmente dabei erst deutlich.

Der letzte Parameter 'Bild' dient dazu, aus Kreisen Kreise zu machen. Sie lachen? Schauen Sie mal auf Ihren Monitor, vielleicht haben Sie sich gewundert, warum der Kreis eher einem Ei gleicht? 'Bild' ist für den Breiten-/Höhenausgleich zuständig. Der Parameter kann einen Wert zwischen 0 und 1 annehmen, beim Start von AmigaBASIC liegt er bei 0.44. Probieren Sie doch mal andere Werte!

Daß man mit Circle auch Ellipsen zeichnen kann, braucht wohl keine nähere Beschreibung, oder? Einfach den Parameter 'Bild' ändern, schon wird's 'ellipsiger'.

Listing 7 zeigt, wie man Kreise auf zwei verschiedene Arten erzeugen kann.

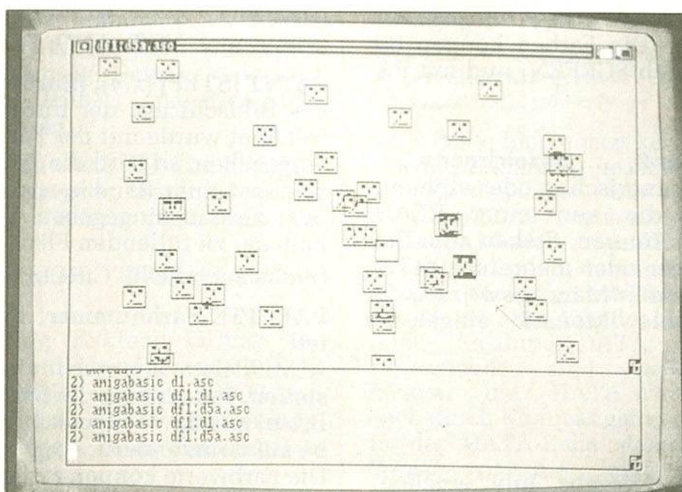


Bild 2: Mit GET und PUT lassen sich Teile eines Bildes herausschneiden und an beliebiger Stelle wieder einsetzen

"Scroll dich", sagte der BASIC-Programmierer – und so geschah es

Mit GET und PUT einen Teilbereich der Grafik herauszukopieren und an anderer Stelle zu positionieren, dies haben wir ja schon kennengelernt. AmigaBASIC gibt uns aber noch einen mächtigen Befehl mit auf den Weg, der es uns erlaubt, bestimmte Bereiche des Bildschirmes zu bewegen, den Befehl SCROLL. Hier die Syntax:

SCROLL(x1,y1)-(x2,y2),delt ax,deltay

Beim SCROLL-Befehl wird zunächst die linke obere Ecke des Bereiches definiert, der gescrollt werden soll (x1,y1)

Das BASIC-Glossar

Begriffe und Befehle des Kurs-Teils

Absolut-Koordinaten – Punkte innerhalb einer Fläche, die sich vom absoluten Nullpunkt berechnen. Beim Bildschirm liegt dieser in der linken, oberen Ecke (x = 0,y = 0). Will man einen Punkt in den Screen setzen, so müssen der x- und der y-Wert um beliebige Pixel in die positive Richtung vom Nullpunkt aus gesetzt werden.
(Siehe auch PIXEL, SCREEN)

AND – Logische Verknüpfung (Konjunktion). Die Bits werden nach folgenden Regeln verknüpft: 0 AND 0 ergibt 0, 1 AND 0 ergibt 0, 0 AND 1 ergibt 0, 1 AND 1 ergibt 1. Damit ist klar, daß nur zwei auf '1' gesetzte Bits auch eine '1' ergeben.
(Siehe auch NOT, OR, XOR, EQV, IMP)

AREA [STEP] (x,y) – Befehl zum Setzen des Eckpunktes eines Polygons (Vielecks). Der Punkt kann absolut (ohne STEP) oder relativ (mit STEP) gesetzt werden.
(Siehe auch AREAFILL, STEP)

AREAFILL [Modus] – Füllt die mit AREA [STEP] erstellten Polygone mit einem Muster, die mit PATTERN vorher definiert wurden. 'Modus' ist optional und braucht nicht angegeben zu werden, dann wird das Polygon mit der Zeichenfarbe ausgefüllt.
(Siehe auch AREA, STEP, PATTERN)

Bogenmass – Dient u.a. zum Berechnen des Kreisumfangs, dabei ist festgelegt, daß ein Halbkreis mit dem Winkel von 180° der Konstante PI (3.1415) entspricht. Also berechnet sich der Vollkreis aus 2 * PI. Gebraucht wird dieses Maß beim CIRCLE-Befehl für Kreisanfang und Kreisende.
(Siehe auch CIRCLE,PI)

CIRCLE [STEP] (x,y), Radius, Farbe, Kreisanfang, Kreisende, Bild – Mit diesem Befehl können Kreise und Ellipsen sowie Kreis- und Ellipsenausschnitte gezeichnet werden. 'x,y' sind entweder relativ oder absolut einzugeben und bilden den Mittelpunkt, 'Radius' wird ebenfalls in Pixeln eingegeben, 'Farbe' kann ein Wert zwischen 0 und 31 (bei entsprechender Screen-Tiefe) und mit PALETTE definiert sein, Kreisanfang und -ende sind Werte in Bogenmaß, 'Bild' steht für Höhen-/Breiten-Verhältnis (entscheidet über Kreis oder Ellipse).
(Siehe auch STEP, PALETTE, Bogenmaß)

COLOR Vordergrund, Hintergrund – Legt die Zeichen- und die Hintergrundfarbe fest. Die Farben können die Nummern 0 bis 31 haben (je nach SCREEN) und mit PALETTE definiert worden sein.
(Siehe auch SCREEN, PALETTE)

DATA Bezeichner 1, Bezeichner 2, ..., Bezeichner n...
In DATA-Zeilen werden die numerischen oder alphanumerischen Werte abgelegt, die von einer READ-Anweisung eingelesen werden können. Stehen zum Beispiel die Monatsnamen in einer oder mehreren DATA-Zeilen (DATA "Januar", "Februar", "März",usw.), so können diese in die Feldvariable 'Monat$' eingelesen werden:

```
DIM Monat$(12)
FOR Anzahl = 1 TO 12
READ Monat$(Anzahl)
NEXT Monat
```

Ein PRINT Monat$(7) würde als Ausgabe "July" ergeben.
(Siehe auch READ)

EQV – Logische Verknüpfung (Äquivalenz). Die Bits werden nach folgenden Regeln verknüpft: 0 EQV 0 ergibt 1, 1 EQV 0 ergibt 0, 0 EQV 1 ergibt 0, 1 EQV 1 ergibt 1. Feste Regel also: Das Ergebnisbit ist dann '1' wenn beide Eingangsbits *gleich* sind.
(Siehe auch NOT, AND, OR, XOR, IMP)

GET (x1,y1)-(x2,y2),Feldvariable% [Index,Index...Index]

– Der GET-Befehl liest einen Teilbereich mit den Maßen 'x1,y1' (linke, obere Ecke) bis 'x2,y2' (rechte, untere Ecke) in die vorher dimensionierte Feldvariable ein. Die Dimensionierung richtet sich nach der Größe des eingelesenen Bereichs (im Text beschrieben!). Mit Index können weitere Feldvariablen genutzt werden.
(Siehe auch PUT)

IMP – Logische Verknüpfung (Implikation). Die Bits werden nach folgenden Regeln verknüpft: 0 IMP 0 ergibt 1, 1 IMP 0 ergibt 0, 0 IMP 1 ergibt 1, 1 IMP 1 ergibt 1. Es gibt hier also nur eine '0' als Ausgangsresultat, wenn Eingangsbit A = '0' und Eingangsbit B = '1' ist. Da dies eine äußerst selten gewünschte Konstellation ist, wird sie auch in BASIC wenig Gebrauch finden.
(Siehe auch NOT, AND, OR, XOR, EQV)

LINE [STEP] (x1,y1)-(x2,y2),[Farbe],[B(lock) oder B(lock)F(ill) – zieht eine Linie von x1,y1 nach x2,y2 in der Farbe 'Farbe' oder ein Rechteck mit der Option 'B' oder ein gefülltes Rechteck mit der Option 'BF'. Die Koordinaten können wieder als absolute oder relative Werte eingegeben werden.
(Siehe auch STEP)

NOT – Logische Operation (Komplement). Jedes gesetzte Bit ('1') wird nach NOT eine '0', jedes nicht gesetzte Bit ('0') wird nach NOT eine '1'. NOT bewirkt bei grafischen Operationen Invertierungen.
(Siehe auch AND, OR, XOR, EQV, IMP)

OR – Logische Verknüpfung (Disjunktion). Für OR gilt die folgende Verknüpfungsvorschrift: 0 OR 0 ergibt 0, 1 OR 0 ergibt 1, 0 OR 1 ergibt 1, 1 OR 1 ergibt 1. Daraus folgt: Eine logische '0' bleibt nur dann übrig, wenn beide Eingangsbits logisch '0' sind.
(Siehe auch NOT, AND, XOR, EQV, IMP)

PAINT [STEP] (x,y), [Farbe],[Rand] – füllt einen Bereich des Bildschirms, der durch LINE oder CIRCLE gekennzeichnet wurde mit der Farbe 'Farbe'. Wird 'Farbe' nicht eingegeben, so wird die Zeichenfarbe genommen. 'Rand' zeichnet einen Rand in anderer Farbe. 'x,y' können relativ oder absolut eingegeben werden, müssen jedoch innerhalb der zu füllenden Fläche liegen.
(Siehe auch LINE, CIRCLE)

PALETTE Farbnummer, Rotanteil, Grünanteil, Blauanteil
– Mit Palette läßt sich für jede Farbnummer eine Farbe erstellen, indem man die Farbanteile von R(ot), G(rün) und B(lau) mischt. Will man zum Beispiel die Hintergrundfarbe auf Grün ändern, so gibt man ein 'PALETTE 0,0,1,0'. Die Farbwerte können zwischen 0 und 1 liegen.
(Siehe auch COLOR)

PATTERN Lmuster,Fmuster – Erstellt ein Füllmuster, wobei 'Lmuster' das Linienmuster, 'Fmuster' das Füllmuster sind.head 'Fmuster' setzt sich aus einer Feldvariablen zusammen, in der das Muster durch Bit-Setzen oder -Löschen entsteht. Weitere Informationen im nächsten Heft. **PI ,, – Konstante zur Kreisberechnung bei CIRCLE. Leider kennt AmigaBASIC die Konstante PI nicht, es reicht jedoch, wenn man PI als Konstante von 3.1415 festlegt.**
(Siehe auch CIRCLE)

Pixel – Kleinstes darzustellendes Grafikelement bei einem Computer (PIcture (X) ELement). Alle Grafikbefehle beziehen sich auf die Pixel.

PSET/PRESET [STEP] (x,y),[Farbe] – dienen beide zum Zeichnen von Pixeln in der Farbe 'Farbe'. Wenn 'Farbe' weggelassen wird, hat der Pixel bei PSET die Zeichenfarbe, bei PRESET die Hintergrundfarbe. 'x,y' können absolut oder relativ gesetzt werden.
(Siehe auch STEP)

PUT [STEP] (x,y), Feldvariable [(Index,Index...Index)], Modus – Der mit GET ausgelesene Bereich kann mit PUT an anderer Stelle ('x,y' für die linke, obere Ecke) wieder eingesetzt werden. Dazu muß die von GET benutzte Feldvariable entweder einer hier genutzten zugewiesen oder übernommen werden, das gleiche gilt für 'Index'. Der Grafikbereich kann mit dem Hintergrund logisch verknüpft werden, und zwar durch AND, OR, XOR. Durch Modus = PSET wird der Bereich genauso gesetzt, wie er übernommen wurde, durch Modus = PRESET wird er invertiert.
(Siehe auch AND, OR, XOR, PSET, PRESET)

READ – dient zum Einlesen von in DATA-Zeilen abgelegten Werten. Diese können numerisch oder Strings (Zahlen wie 1,2,65,282882, Wörter wie "Januar","Hallo Welt","AMIGA DOS") sein. Auch in BASIC implementierte Maschinenprogramme können so in ihren Speicherbereich eingelesen werden.
(Siehe auch DATA)

Relativ-Koordinaten – beziehen sich immer auf den letzten aktuellen Punkt und erklären diesen zum Nullpunkt. War der letzte Punkt zum Beispiel bei x = 20 und y = 20, so werden die neuen Koordinaten bei positiver Richtung zugezählt, bei negativer Richtung abgezogen.
(Siehe auch Absolut-Koordinaten)

SCROLL (x1,y1)-(x2,y2),DeltaX,DeltaY – Scrollt (kommt von SCreen ROLLing (Bildschirmrollen)) einen Teilbereich des Screens mit dem Anfang 'x1,y1' (linke, obere Ecke) und 'x2,y2' (rechte, untere Ecke) um DeltaX Pixel (-DeltaX = negative Richtung, DeltaX = positive Richtung) und DeltaY Pixel (-DeltaY = negative Richtung, DeltaY = positive Richtung).

STEP – kann bei Grafikbefehlen wie PSET und PRESET, LINE, CIRCLE und AREA eingesetzt werden, um die Koordinaten nicht absolut, sondern relativ zu berechnen.
(Siehe auch 'Absolut-Koordinaten, 'Relativ-Koordinaten')

XOR – Logische Verknüpfung (Exklusiv ODER). Die Verknüpfungen zwischen zwei Bits sehen so aus: 0 XOR 0 ergibt 1, 1 XOR 0 ergibt 0, 0 XOR 1 ergibt 0, 1 XOR 1 ergibt 1. Damit steht fest, daß das Verknüpfungsergebnis nur dann '1' ist, wenn beide Eingangsbits gleich sind.
(Siehe auch NOT, AND, OR, EQV, IMP)

– dies sind absolute Bildschirmkoordinaten. Anschließend wird die rechte untere Ecke des Bildschirmbereichs mit den Koordinaten x2 und y2 festgelegt. Fehlen nun noch die Werte, um die unser Bildchen bewegt werden soll. Deltax ist ein numerischer Ausdruck, dessen Wert die Anzahl der Bildpunkte angibt, um die entweder nach links oder rechts gescrollt werden soll. Die Werte sind jeweils ganze Zahlen, die entweder positiv – in diesem Fall bewegt sich das Bild nach rechts – oder negativ (Bild bewegt sich nach links) gesetzt werden. Da das Bildchen aber nicht nur nach links und rechts, sondern auch noch nach unten bzw. oben bewegt werden soll, müssen noch einige Werte her. Dazu schreiben wir in deltay einen positiven bzw. negativen ganzzahligen Wert hinein. Der positive Wert steht für ein Scrollen des Bildschirms nach unten und dementsprechend rollt er nach oben, wenn's ein negativer Wert ist. So, jetzt haben wir das grundsätzliche Gerüst für den Umgang mit SCROLL, um uns das Ganze anhand eines kleinen Beispiels zu verdeutlichen:

```
WINDOW 1:CLS
FOR lesen = 1 TO 3
  READ text$
  LOCATE 10,10:PRINT text$
  LINE(60,70)-(350,80),2,b
  FOR n=1 TO 2000:NEXT n
  WHILE zaehler < 20
     SCROLL(61,71)-(349,79),
0,1
     zaehler = zaehler + 1
  WEND
  zaehler = 0
NEXT lesen
END
DATA "Der SCROLL-Befehl macht
folgendes:"
DATA "Ein Teilbereich des Bil
des wird ge"DATA"scrollt. Ist
doch schon toll, oder?"
```

Ich möchte hierzu jetzt keine nähere Erklärung machen, nur für diejenigen, die mit READ und DATA noch nichts anfangen können, folgendes:
Mit READ kann man eine bestimmte Menge an Variablen oder Daten nacheinander einlesen, die durch DATA gekennzeichnet sind. Dabei können die DATA-Werte auch durch Kommas getrennt in der DATA-Zeile stehen. Wenn wir die folgende DATA-Anweisung DATA 1,2,3,4,5 mit 'READ a:PRINT a' einlesen (in einer Schleife, wohlgemerkt!), so erhalten wir nacheinander für 'a' die Werte 1,2,3,4,5. Auf diese Weise kann man auch Feldvariablen 'füttern', und dies sogar mit Stringvariablen, also Text (wie in unserem Beispiel). Mit READ und DATA sollte man sich als BASIC-Neueinsteiger ruhig einmal näher befassen.
Im Beispiel sollten Sie einmal die letzten beiden Werte vertauschen, dann muß noch etwas anderes verändert werden, aber dieses sollten Sie selbst herausfinden.
So, das waren die etwas schwierigeren Befehle, kommen wir jetzt zum Rest des Grafikbestandes.

Soll das Werk den Meister loben, kommt die Farbe ins Spiel

Immer nur die drei Grundfarben zu benutzen ist auch nicht das Wahre, zumal, wenn Sie immer noch die voreingestellten Farben Blau, Orange, Weiß und Schwarz benutzen. Hätten Sie's nicht gern auch mal in Rot oder Grün, Lila wär' ja auch nicht zu verachten. Nun gibt es da einen ganz einfachen Befehl, um die Farben des Hintergrundes (Screen-Fläche) und des Vordergrundes (Schrift) zu verändern und dieser Befehl heißt 'COLOR Vordergrund, Hintergrund'. Mit COLOR 1,3 ändern wir also die aktuellen Farben, aber nur in der Zeile, in der wir uns befinden. Setzen wir hinter den COLOR-Befehl noch ein CLS (CLear Screen), so wird der gesamte Bildschirm in der aktuellen Farbe 'gelöscht'. Versuchen wir es:

```
COLOR 0,1:CLS o.k.
COLOR 3,2:CLS o.k.
COLOR 2,0:CLS o.k.
COLOR 2,4:CLS
```

Das war wohl nichts! Aber warum? Erinnern wir uns noch einmal an den SCREEN-Befehl. Jeder neue Screen, der in AmigaBASIC geöffnet wird, hat spezielle Eigenschaften, darunter die sogenannte Tiefe. Sie gibt an, wie viele Farben wir verwenden können. Unser vordefinierter Screen kann nur die vier Grundfarben (0 bis 3) annehmen. War also nichts mit Farbe, oder?
Der Trick liegt darin, daß uns der Amiga selbst entscheiden läßt, welche Farbe wir aus der

Palette der 4096 nehmen. Und das ist das Geheimnis. Schauen wir noch mal auf den Screen: 'Tiefe' gab an, wie viele Farben nutzbar für uns sind. Nehmen wir doch mal einen Screen mit der Tiefe 4 und der Größe 320*256 Pixel (wir erinnern uns, unser PAL-Amiga kann das). Schauen wir nun noch einmal in die letzte AMIGA DOS, so sehen wir, daß ein Screen mit der Tiefe 4 insgesamt 16 Farben verwalten kann. Geben wir also ein

```
SCREEN 2,320,256,4,1
WINDOW 2,"",2:'Reicht a
ls Alibi-Window
```

O.k., das hätten wir, jetzt noch die Farben. Als nächstes geben wir ein:

```
FOR n=0 TO 15
  LOCATE n+1,1
  COLOR n,0:PRINT"Hallo W
elt"
  NEXT n
  WHILE INKEY$="":WEND
  WINDOW CLOSE 2:SCREEN CLOS
E 2
```

Nach dem Start grüßen wir die Welt in 16 Farben. Sie sehen nur 15? Na klar, Schrift in Farbe 0, Hintergrund in Farbe 0, was wollen Sie da sehen! Schön, wir haben jetzt 16 Farben, aber eigentlich wollten wir doch Mitspracherecht über deren Aussehen haben? Nichts einfacher als das! Merken Sie sich doch mal die Farbe von 'Hallo Welt' Nummer sieben. Sie hat die Farbnummer 6 (0 bis 6 sind sieben, zählen Sie nach). Gehen Sie jetzt in das Programm, und geben Sie eine neue Zeile unter dem WINDOW-Befehl ein:

```
PALETTE 6,0,1,0
```

Nach erneutem Starten des Programms vergleichen Sie die Farbe mit der von vorhin. 'Is nich mehr', wie der gebildete Fachmann sagt. Und schon kennen wir einen neuen Befehl.

PALETTE Farbnummer, Rotanteil, Grünanteil, Blauanteil heißt er, und in ihm findet man das Geheimnis der vielen Farben. Farbnummer ist die Farbe, die geändert werden soll, Rot, Grün, Blau sind die Farbwerte für diese drei Farben. Sie kann man in Abstufungen von 0 bis 1 ändern. Am besten auch hierzu ein Beispiel:

Mit PALETTE 3,1,0,0 ändern wir unsere Grundfarbe 3 (die bei jedem von Ihnen anders aussehen kann, angefangen von der 'Startfarbe' bis hin zu durch 'Preferences' erstellte Farben) in ein kräftiges Rot. Wenn wir aus diesem Rot ein Orange oder gar ein Gelb machen wollen, muß eine gehörige Portion Grün in den Farbeimer. Also flugs eingegeben 'PALETTE 3,1,0.55,0 und schon wird es orange. Nach PALETTE 3,1,1,0 wird es hellgelb. Versuchen Sie es einmal, und überlegen Sie sich, welche Mischung wohl ein Dunkelgrün hervorbringt. Nachdem wir jetzt alles über Farben und Flächen, Linien, Kreise und Punkte wissen, gehen wir die 'letzten' Grafikbefehle an (die letzten sind es noch nicht, aber dazu später mehr!). Da wäre zuerst einmal PAINT. Die Syntax dazu

PAINT [STEP] (xKoordinate, yKoordinate),Farbe,[Rand]

PAINT dient dazu, eine umrandete Fläche mit einer Farbe auszumalen, zu füllen. Um den Rand kenntlich zu machen, kann eine RAND-Farbe mit angegeben werden. Die Koordinaten x/y können wieder relativ oder absolut eingegeben werden.

AREA [STEP] (x,y) dient zum Zeichnen von Polygonen (Vielecken), auch hier gibt es die STEP-Funktion. **AREAFILL [Modus]** füllt dieses Polygon mit einem Muster.

Schön, nun kennen wir die restlichen Befehle, aber was wäre ein Kurs ohne Beispiele? Also fix was ausgedacht:

Mit AREA können wir Punkte definieren, die ein Vieleck markieren, mit AREAFILL können wir dieses füllen. Versuchen wir es:

```
'Der weiße Hai (Teil 94)'
WINDOW 1
PALETTE 1,0,0,0:PALETTE 3,1,0
0
COLOR 1,3:CLS
FOR n=1 TO 600 STEP 150
  AREA (n,1)
  AREA (n+100,1)
  AREA (n+50,120)
  AREAFILL
  IF n < 400 THEN
     AREA (n+75,180)
     AREA (n+175),180
     AREA (n+125,60)
     AREAFILL
  END IF
NEXT n
WHILE INKEY$="":WEND
```

Das Ergebnis... Nun, sehen Sie selbst!

Was ist, wenn wir anstelle der AREAFILL-Anweisung die PAINT-Anweisung setzen? Nun, ganz einfach, dann wird der gesamte Bildschirm in der PAINT-Farbe gefüllt. Der Grund dafür ist klar: Mit AREA lassen sich nur die Punkte eines Polygons setzen, und AREAFILL füllt dieses Polygon. PAINT braucht jedoch einen geschlossenen Körper, der vorher mit LINE oder CIRCLE erzeugt werden muß:

```
WINDOW 1:CLS
CIRCLE (320,100),90,3
PAINT (320,100),1,3
```

erzeugt einen in Farbe 1 gefüllten Kreis, der einen durch Farbe 3 gekennzeichneten Rand hat. Vorsicht, ist der Kreis offen, stimmen also Rand- und Füllfarbe überein, kann es passieren, daß nicht nur der Kreis, sondern der gesamte Bildschirm gefüllt wird.

Kommen wir jetzt zum letzten Rätsel für dieses Mal, dem MODUS für AREAFILL. MODUS erlaubt den Einsatz eines Füllmusters, welches mit PATTERN erstellt werden kann. MODUS kennt dabei zwei Zustände:

0 -> Der PATTERN-Wert ist voreingestellt und hat den Wert der Zeichenfarbe,

1 -> der PATTERN-Wert hat den Wert der Hintergrundfarbe.

Bei Fall zwei stellt sich eine Invertierung des ersten Falles dar.

Die Sache mit den FD-Dateien

Nach den ersten Folgen unseres BASIC-Kurses hatten viele Einsteiger Fragen über diese mitgelieferte Sprache des Amiga. Eine der meistgestellten Fragen betrifft das Verzeichnis FD1.3 (FD1.2) auf der mitgelieferten ExtrasD-Diskette, bei dem man nicht so genau weiß, was man damit machen soll. Die darin enthaltenen Dateien sind weder lad- noch ausführbar, bringen also dem BASIC-Programmierer keinen direkten Nutzen. Oder doch?

Bei den FD-Dateien handelt es sich um Bibliotheks-Dateien des Amiga, also Dateien, die direkt etwas mit dem Betriebssystem des Rechners zu tun haben. Diese FD-Dateien liegen jedoch nicht direkt gebrauchsfertig vor. Sie müssen erst in ein neues Format umgewandelt werden, nämlich in die sogenannten '.BMAP'-Dateien. Einige der '.BMAP'-Dateien sind allerdings schon fertig umgewandelt, sie befinden sich jedoch wieder ganz woanders – nämlich im BASICDEMOS-Verzeichnis. Hier finden wir die 'diskfont.bmap', die 'dos.bmap', die 'exec.bmap' und die 'graphics.bmap'. Diese '.BMAP'-Dateien sind für Einsteiger nicht ganz leicht zu handhaben, dazu gehört schon ein bißchen Grundwissen über das System Amiga an sich, deshalb sollen Sie auch hier und jetzt nicht behandelt werden. Wir werden aber später über dieses Thema noch ausführlich schreiben, da gerade hier AmigaBASIC einige seiner Schwachpunkte wieder aufheben kann, da die '.BMAP'-Dateien eine systemnahe Programmierung zulassen.

Im BASICDEMOS-Verzeichnis finden wir auch das Umwandlungsprogramm für die FD-Dateien, genannt 'ConvertFD'. Es handelt sich dabei um ein ganz normales BASIC-Programm, welches vom Editor geladen und ausgeführt werden kann. Die entsprechende FD-Datei muß mit Pfadverzeichnis angegeben werden und wird dann automatisch in die '.BMAP'-Datei konvertiert. Zu diesem Zweck sollte man auf seiner BASIC-Diskette ein Verzeichnis namens 'BMAPS' einrichten, da die meisten, in BASIC geschriebenen Programme, wenn sie Gebrauch von den '.BMAP'-Dateien machen, dieses Directory suchen. Auch Amiga-BASIC-Programme, die in der AMIGA DOS erscheinen, können die '.BMAP'-Dateien benutzen, deshalb ist es sicherer, sich dieses Verzeichnis vorher einzurichten.

Die in BASICDEMOS schon fertig vorhandenen '.BMAP'-Dateien sollten Sie daher schon in das 'BMAPS'-Verzeichnis kopieren, Benutzer zweier Laufwerke haben da keine Probleme, diejenigen, die nur ein Laufwerk haben, müssen hier den Umweg über die RAM-Disk gehen. Das Ganze sieht für Besitzer eines Laufwerks so aus:

1) Verzeichnis einrichten mit MakeDir,
2) '.BMAP'-Dateien aus dem BASICDEMOS-Verzeichnis der ExtrasD-1.3 (1.2)-Diskette in die RAM-Disk kopieren.
3) Von der RAM-Disk die '.BMAP'-Dateien in das neue Verzeichnis kopieren.

Damit dürften alle Amiga-BASIC-Programme, die Sie aus unserer oder anderen Zeitschriften abtippen und die auf die '.BMAP'-Dateien zugreifen, lauffähig sein.

Und was man mit diesen '.BMAP'-Dateien alles anstellen kann, das zeigen wir Ihnen in einer der nächsten Ausgaben.

Um an ein Füllmuster zu gelangen, müssen wir dieses erst mit der PATTERN-Anweisung erstellen, die Syntax lautet:

PATTERN,Lmuster,FMuster

Während 'Lmuster' einen 16-Bit-Wert repräsentiert (voreingestellt ist die Zahl 65535), basiert 'Fmuster' auf eine vorher definierte Feldvariable. In dieser können wir die Höhe des Füllmusters in Bildpunkten bestimmen. Die Anzahl der Feldelemente ist dabei eine Zweierpotenz, also 1,2,4,8,16, usw. Gesetzte Bits und nichtgesetzte Bits sind hier wieder die Zauberworte. Jedes gesetzte Bit wird gezeichnet, jedes nichtgesetzte ausgelassen.

Um den PATTERN-Befehl zu verstehen, braucht es etwas mehr Platz, wir werden uns deshalb im nächsten Teil des Kurses noch einmal mit ihm beschäftigen müssen. Eines jedoch vorweg: Für wen jetzt noch Bits und Bytes böhmische Dörfer sind, der sollte sich schleunigst damit befassen, denn gerade in der Grafik basiert vieles auf '1' oder '0'.

Die Grafik unter AmigaBASIC ist damit noch lange nicht ausgereizt, jedoch soll dies fürs erste genügen. Wir werden uns später einmal mit den 'Libraries', den Bibliotheken befassen, zu denen auch die 'graphics.library' gehört, mit der auch Textspiegelung und vieles mehr möglich ist. Die 'graphics.library' ist eigentlich schon ein Einstieg in die Tiefen des Betriebssystem unseres Amiga und nicht gerade leicht zu handhaben. Für unsere Einsteiger soll daher erst einmal hiermit genug sein.

Für heute ist es genug mit dem Lernen, versuchen Sie es jetzt mal selbst!

Auch von Sprite- und BOB-Programmierung (Blitter-OBjects) soll hier noch nicht die Rede sein, auch wenn AmigaBASIC eine Fülle an OBJECT-Befehlen kennt. Mit den eben vorgestellten Grafikbefehlen läßt sich doch schon eine Menge anfangen, oder?

Die Aufgabe:

Ach ja, wissen Sie noch? Am Anfang dieses Kursteils haben wir vom 'Mondgesicht' gesprochen. Wie wär's? Zeichnen Sie sich doch selbst einmal eines. Und zwar eines, welches als Grundform ein Sechseck hat, und jedes Eckteil soll eine andere Farbe haben. Damit es noch ein bißchen schwieriger wird: Hier sind die Farben:

– ROT, GRÜN, BLAU, GELB, ROSA, BRAUN.

Und Augen soll es auch haben, kreisrund, und einen Mund aus zwei Kreisteilen. Nicht zu vergessen die dreieckige Nase. Nun, schaffen Sie es?

Im Anhang finden Sie noch den zweiten Teil unseres BASIC-Glossars mit den vorgestellten Befehlen (falls Sie mal nicht weiterwissen).

Bis zum nächsten Mal!

(jb/vb)

Listings

```
000:023 '             demo zum
001:004 '      oeffnen und schliessen
002:725 '     von screens und windows
003:000
004:398  SCREEN 2,640,256,3,2
005:918  WINDOW 1,"window 1",(1,1)-(600,200),3,2
006:012   PRINT"hallo welt !"
007:806  WINDOW 2,"window 2",(20,30)-(300,100),3,2
008:012   PRINT"hallo welt !"
009:918  WINDOW 3,"window 3",(50,80)-(400,150),3,2
010:012   PRINT"hallo welt !"
011:000
012:977 FOR w=1 TO 3
013:259  WINDOW OUTPUT w
014:379  FOR i=200 TO 0 STEP -1
015:361   LOCATE 1,20:PRINT  i
016:393  NEXT i
017:501  WINDOW CLOSE w
018:407 NEXT w
019:000
020:576 SCREEN CLOSE 2
```

```
000:240 '  demo zum pset befehl
001:350 '  "sternenhimmel"
002:000
003:503 CLS
004:000
005:849 DIM x (500) , y (500)
006:000
007:268 FOR i = 1 TO 500
008:337  x = INT ( RND (1) * 640 )
009:825  y = INT ( RND (1) * 200 )
010:414  x(i) = x
011:447  y(i) = y
012:925  PSET ( x , y ) , 1
013:393 NEXT i
014:212 m:
015:268 FOR i = 1 TO 500
016:565  PSET ( x(i) , y(i) ) , a
017:393 NEXT i
018:983 d = d + 1
019:635 a = ABS ( a < 1 )
020:547 IF d < 5 THEN GOTO m
021:569 ERASE x,y
```

```
000:256 ' demo zum line-befehl
001:268 ' bewegen  von  linien
002:000
003:074 Start:
004:270    x1=x2
005:288    y1=y2
006:448    x2=RND(1)*600
007:528    y2=RND(1)*200
008:000
009:288    x3=x4
010:306    y3=y4
011:544    x4=RND(1)*600
012:624    y4=RND(1)*200
013:000
014:874    schritte = 30
015:000
016:547    DIM x1(schritte),x2(schritte),y1(schritte),y2(sc
        hritte)
017:000
018:885    gx1=(x2-x1)/schritte
019:461    gy1=(y2-y1)/schritte
020:389    gx2=(x4-x3)/schritte
021:965    gy2=(y4-y3)/schritte
022:000
023:845    FOR i= 0 TO schritte
024:669       x1(i)=x1+i*gx1
025:991       y1(i)=y1+i*gy1
026:894       x2(i)=x3+i*gx2
027:216       y2(i)=y3+i*gy2
028:160    NEXT
029:519    FOR i=0 TO schritte-1
030:429       LINE (x1(i+1),y1(i+1))-(x2(i+1),y2(i+1)),1
031:916       LINE (x1(i),y1(i))-(x2(i),y2(i)),0
032:160    NEXT
033:796    ERASE x1,x2,y1,y2
034:700    GOTO Start
```

```
000:334 REM *** worms  ***
001:000
002:240    INPUT"linker  rand :";lr  : IF lr =0 THEN lr=200
003:888    INPUT"rechter rand :";rr  : IF rr =0 THEN rr=400
004:270    INPUT"oberer  rand :";obr : IF obr=0 THEN obr=50
005:310    INPUT"unterer rand :";ur  : IF ur =0 THEN ur=150
006:595    INPUT"schrittweite :";a   : IF a = 0 THEN a=5
007:503    CLS
008:000
009:798    LINE (lr,obr)-(rr,ur),1,b
010:000
011:556    mx=lr+(rr-lr)/2
012:684    my=obr+(ur-obr)/2
013:862    x1=mx:x2=mx:x3=mx:x4=mx
014:047    y1=my:y2=my:y3=my:y4=my
015:622    x5=mx:x6=mx:x7=mx:x8=mx
016:807    y5=my:y6=my:y7=my:y8=my
017:606    x9=mx:x10=mx:x11=mx:x12=mx
018:119    y9=my:y10=my:y11=my:y12=my
019:910    x13=mx:x14=mx:x15=mx:x16=mx
020:855    y13=my:y14=my:y15=my:y16=my
021:000
022:190    zhoriz1=0
023:207    zvert1=-1
024:194    zhoriz2=0
025:587    zvert2=1
026:441    zhoriz3=-3
027:590    zvert3=0
028:205    zhoriz4=3
029:594    zvert4=0
030:000
031:074 Start:
032:445    LINE (x1,y1)-(x2,y2),1
033:874    LINE (x4,y4)-(x3,y3),0,bf
034:461    LINE (x5,y5)-(x6,y6),1
035:002    LINE (x8,y8)-(x7,y7),0,bf
036:493    LINE (x9,y9)-(x10,y10),1
037:058    LINE (x12,y12)-(x11,y11),0,bf
038:133    LINE (x13,y13)-(x14,y14),1
039:858    LINE (x16,y16)-(x15,y15),0,bf
040:000
```

```
041:641     x=a/2-INT(RND(1)*a)+zhoriz1
042:401     y=a/2-INT(RND(2)*a)+zvert1
043:595     x4=x3:y4=y3:x3=x2:y3=y2:y2=y1:x2=x1:x1=x1+x:y1=y
1+y
044:584     IF x1>rr-1 THEN
045:299                   x1=rr-1
046:425                   zhoriz1=-3
047:091                 ELSE
048:097                   IF x1<lr+1 THEN  x1=lr+1:zhoriz1=3
049:216     END IF
050:000
051:992     IF y1>ur-1 THEN
052:387                   y1=ur-1
053:208                   zvert1=-2
054:091                 ELSE
055:271                   IF y1<obr+1 THEN  y1=obr+1:zvert1=
1
056:216     END IF
057:000
058:219  ' ========================  worm 2  ============
059:834     x=a/2-INT(RND(3)*a)+zhoriz2
060:498     y=a/2-INT(RND(4)*a)+zvert2
061:723     x8=x7:y8=y7:x7=x6:y7=y6:y6=y5:x6=x5:x5=x5+x:y5=y
5+y
062:632     IF x5>rr-1 THEN
063:427                   x5=rr-1
064:433                   zhoriz2=-3
065:091                 ELSE
066:629                   IF x5<lr+1 THEN  x5=lr+1:zhoriz2=3
067:216     END IF
068:000
069:040     IF y5>ur-1 THEN
070:515                   y5=ur-1
071:216                   zvert2=-2
072:091                 ELSE
073:259                   IF y5<obr+1 THEN  y5=obr+1:zvert2=
1
074:216     END IF
075:000
076:315  ' ========================  worm 3  =============
==
077:643     x=a/2-INT(RND(1)*a)+zhoriz3
078:403     y=a/2-INT(RND(2)*a)+zvert3
079:779     x12=x11:y12=y11:x11=x10:y11=y10:y10=y9:x10=x9:x9
=x9+x:y9=y9+y
080:680     IF x9>rr-1 THEN
081:555                   x9=rr-1
082:441                   zhoriz3=-3
083:091                 ELSE
084:161                   IF x9<lr+1 THEN  x9=lr+1:zhoriz3=3
085:216     END IF
086:000
087:088     IF y9>ur-1 THEN
088:643                   y9=ur-1
089:224                   zvert3=-2
090:091                 ELSE
091:247                   IF y9<obr+1 THEN  y9=obr+1:zvert3=
1
092:216     END IF
093:000
094:851  ' ========================  worm 4  =============
==
095:000
096:836     x=a/2-INT(RND(3)*a)+zhoriz4
097:500     y=a/2-INT(RND(4)*a)+zvert4
098:035     x16=x15:y16=y15:x15=x14:y15=y14:y14=y13:x14=x13:
x13=x13+x:y13=y13+y
099:264     IF x13>rr-1 THEN
100:131                   x13=rr-1
101:449                   zhoriz4=-3
102:091                 ELSE
103:277                   IF x13<lr+1 THEN  x13=lr+1:zhoriz
4=3
104:216     END IF
105:000
106:696     IF y13>ur-1 THEN
107:283                   y13=ur-1
108:232                   zvert4=-2
109:091                 ELSE
110:059                   IF y13<obr+1 THEN  y13=obr+1:zver
t4=1
111:216     END IF
112:000
113:700     GOTO Start
```

```
000:343  '        face 1
001:372  '   malen von 10*100
002:734  '      gesichtern
003:000
004:102     FOR i=1 TO 10
005:380        FOR j=1 TO 100
006:750           LINE(28+x,2+y)-(53+x,14+y),1,b
007:898           LINE(33+x,4+y)-(37+x,5+y),1,bf
008:714           LINE(45+x,4+y)-(49+x,5+y),1,bf
009:085           LINE(41+x,5+y)-(41+x,6+y),1
010:634           LINE(40+x,7+y)-(42+x,8+y),1,bf
011:666           LINE(33+x,10+y)-(49+x,11+y),0,b
012:805           LINE(33+x,12+y)-(49+x,12+y),1
013:001           FOR d=0 TO 10 STEP 5
014:069              LINE(35+i+x,10+y)-(37+i+x,10+y),1
015:725              PSET(36+i+x,11+y),1
016:708              LINE(35+i+x,12+y)-(37+i+x,12+y),0
017:388           NEXT d
018:826           x=RND(1)*530
019:868           y=RND(1)*180
020:394        NEXT j
021:503        CLS
022:393     NEXT i
```

```
000:798  gesicht2:
001:073     DIM a%(200)
002:503     CLS
003:306     LINE(9,2)-(33,15),1,b          :REM kopf
004:039     LINE(13,4)-(17,5),1,bf         :REM auge
005:615     LINE(25,4)-(29,5),1,bf         :REM auge
006:791     LINE(21,5)-(21,6),1            :REM nase
007:458     LINE(20,7)-(22,8),1,bf
008:516     LINE(13,10)-(29,13),1,b        :REM mund
009:413     LINE(13,12)-(29,12),1
010:241     FOR i=0 TO 10 STEP 5
011:109        LINE(15+i,10)-(17+i,10),1
012:245        PSET(16+i,11),1
013:508        LINE(15+i,12)-(17+i,12),0
014:160     NEXT
015:000
016:751     GET (8,1)-(34,16),a%
017:000
018:102     FOR i=1 TO 10
019:380        FOR j=1 TO 100
020:321           x=INT(RND(1)*600)
021:849           y=INT(RND(1)*180)
022:199           PUT (x,y),a%
023:394        NEXT j
024:503        CLS
025:393     NEXT i
```

```
000:023  '    demo zum
001:264  '  circle befehl
002:074  Start:
003:503     CLS
004:553     FOR i=1 TO 100 STEP 5
005:389        CIRCLE (320,100),i,1
006:393     NEXT i
007:089     FOR i=1 TO 2000:NEXT i
008:503        CLS
009:434        b = 0
010:610        WHILE b<.6
011:074           CIRCLE (320,100),100,1,,,b
012:909           b=b+.05
013:196        WEND
014:089        FOR i=1 TO 2000:NEXT i
015:700     GOTO Start
```

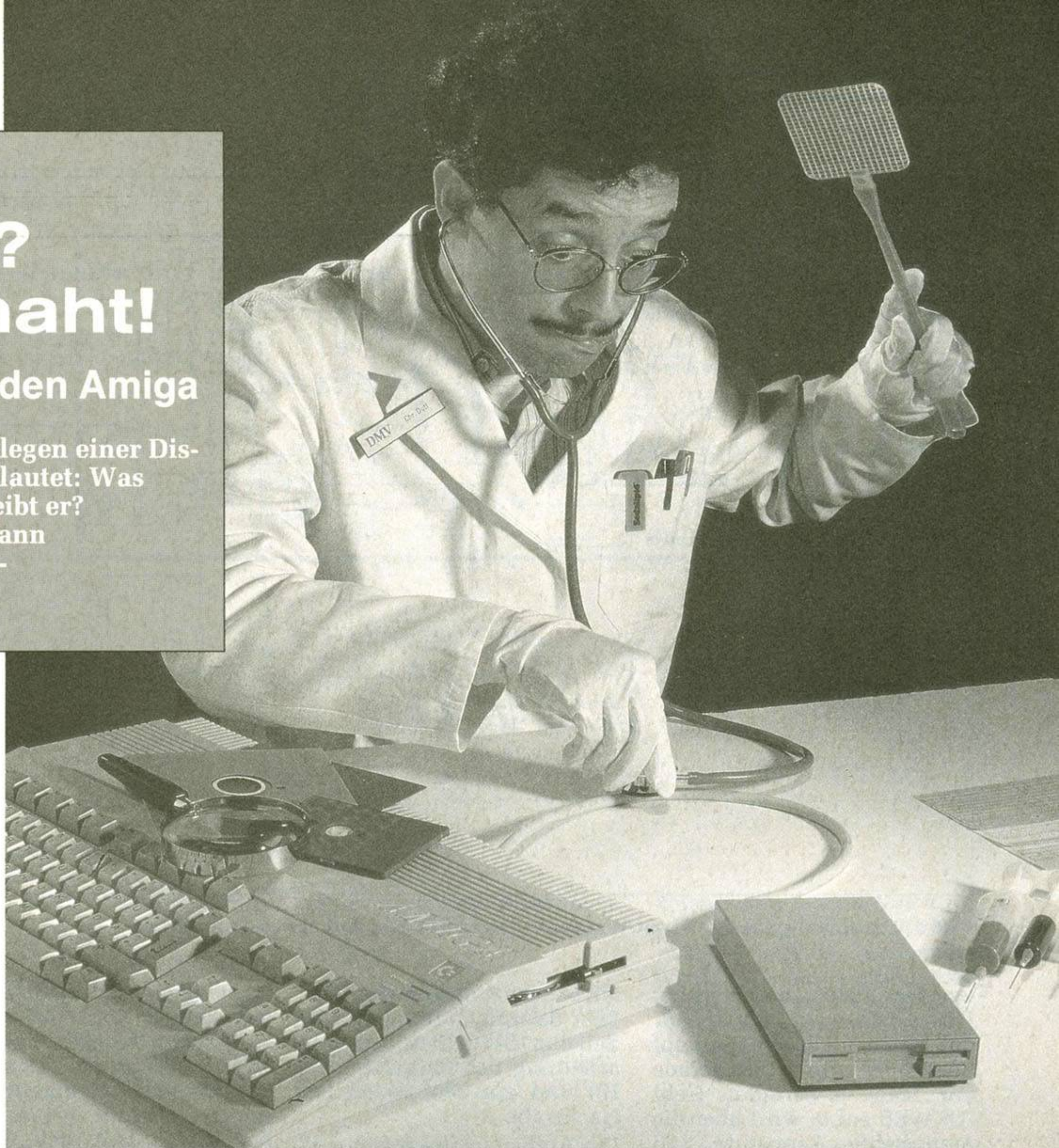
```
000:023  '        demo zum
001:408  '     scroll befehl
002:039  '
003:975  ' bildaufbau
004:000
005:398     SCREEN 2,640,256,3,2
006:254     WINDOW 2,"window 2",(10,10)-(330,195),3,2
007:000
008:116     a$="Amiga-Basic"
009:038     PRINT a$
010:090     PRINT "hat's drauf"
011:000
012:158     LINE (0,161)-(311,161),1,b
013:614     LINE (311,0)-(311,161),1,b
014:000
015:108     FOR i=1 TO 40
016:068        dx=2
017:489        SCROLL (0,0)-(310,150),dx,dy
018:393     NEXT i
019:000
020:994     LOCATE 22,10:PRINT "Taste druecken"
021:000
022:751     pi=3.14159
023:276     dx=4: x=80
024:000
025:074  Start:
026:034     taste$ = INKEY$
027:212     FOR i=0 TO pi/2 STEP pi/40
028:892        x=x+dx
029:307        dy=SIN(i)*11
030:812        IF x+8*LEN(a$) > 310 OR x<10 THEN dx= -dx
031:745        SCROLL (0,0)-(310,160),dx,dy
032:393     NEXT i
033:037     IF taste$<>"" THEN GOTO Ende
034:180     FOR i=3*pi/2 TO 2* pi STEP pi/40
035:892        x=x+dx
036:307        dy=SIN(i)*11
037:812        IF x+8*LEN(a$) > 310 OR x<10 THEN dx= -dx
038:745        SCROLL (0,0)-(310,160),dx,dy
039:393     NEXT i
040:700     GOTO Start
041:196  Ende:
042:174     dx=2:dy=0
043:262     FOR i=1 TO 150
044:745        SCROLL (0,0)-(310,160),dx,dy
045:393     NEXT i
046:503     CLS
047:819     LOCATE 10,11
048:936     PRINT  "schoen was ?"
049:625     FOR i=1 TO 5000: NEXT i
050:464     WINDOW CLOSE 2
051:576     SCREEN CLOSE 2
```

Michael Anton

Hört Ihr es? Der Virus naht!

Schreibindikator für den Amiga

Die bange Frage bei jedem Einlegen einer Diskette in den laufenden Amiga lautet: Was macht er – liest er, oder schreibt er? Ein kleiner Hardwarezusatz kann hier nicht nur endgültige Klarheit verschaffen, sondern auch Daten retten.



Ein Virus hat's nicht leicht, legt der Normalanwender doch absolut keinen Wert auf sein Vorhandensein und versucht, ihn mit allen Mitteln auszuschalten. Klar, daß man sich als Virus dann möglichst unauffällig verhält und dennoch jede Möglichkeit zur weiteren Verbreitung ausnutzt. Von Vorteil ist, daß der Anwender nicht genau weiß, was mit der gerade eingelegten Diskette so alles passiert. Daß das Laufwerk etwas schnurrt, ist ja ganz normal, schließlich muß die Scheibe ja angemeldet werden. Dazu reicht zwar ein Lesevorgang, aber warum soll man als Virus nicht ausnützen, daß die Dinge gerade in Bewegung geraten sind? Und ein bisserl schreiben hat ja noch niemandem geschadet, also drauf mit dem Abkömmling.

Das Opfer Anwender kann sich nur selten mit dem Überlebenstrieb des Virus und den perversen Trieben des Virenprogrammierers identifizieren, insbesondere dann nicht, wenn der Kopierschutz des einzigen Originals in der Sammlung oder der eigene Rache-Virus soeben zerschossen wurden. Zuverlässig schützt nur der Schreibschutz auf der Diskette, aber irgendwann möchte man seine Daten ja auch ablegen – sehr zur Freude aller Viren. Denn dann macht es 'Zisch', und wieder wurde ein Lamer exterminiert. Das Dumme ist nur, daß man dieses Zischen eben nicht hört, mit dem auf Diskette geschrieben wird. Die Wahrheit erfährt man erst, wenn nichts mehr geht, oder bei der routinemäßigen Kontrolle des Diskettenbestandes. Resetfeste Viren können diesen so gut wie unbemerkt verseuchen, wenn man ihnen nicht sofort auf die Spur kommt. Etwas Hardware kann hierbei hilfreich sein.

Piepst es, ist's zu spät...

Die folgende kleine Schaltung, die an den Floppyport des Amiga angeschlossen wird, kommentiert jeden Schreibzugriff auf eines der Diskettenlaufwerke sowohl optisch als auch akustisch. Für die gerade eingelegte Diskette erfolgt die Warnung zwar zu spät, aber immer noch früh genug, um andere zu retten (und vielleicht macht man sich's ja zur Angewohnheit, immer wieder mal eine 'Probediskette' einzulegen...). Da Luxus nicht gescheut wurde, ist auch eine genauere Lokalisierung des Schreibvorgangs vorgesehen: Wird auf Track 0 geschrieben, also in den Bereich, in dem auch der Bootblock liegt, erfolgt die Warnung entsprechend intensiver, durch schnelleres Blinken und Piepsen (die Unterschiede sind am besten bei der Formatierung einer Diskette erkennbar).

Das Schreiben auf Diskette (nicht auf RAM- oder Harddisk!) wird nun sicher erkannt, die Legalität des Vorgangs muß jedoch der Anwender selbst beurteilen. Das sicherste Indiz für einen Virus ist das Schreiben unmittelbar nach dem Einlegen einer neuen Diskette, bevorzugt auf Track 0. Während des Bootvorgangs ist ein Schreiben normalerweise ungewöhnlich, muß aber kein Indiz für einen Virus sein. Ein Blick auf die Startup-Sequence kann hier Klarheit verschaffen. Die Ursachen für ein Schreiben während der normalen Arbeit sind allerdings nur schwer zu erkennen. Vom Anwender getätigte Operationen, wie das Speichern eines Textes, sind noch sonnenklar, bei plötzlichen Warnungen ist jedoch nicht immer Grund zur Aufregung geboten. So kann es durchaus sein, daß eine Anwendung einmal Daten ablegen muß und dies auf Diskette tut. Auch das Betriebssystem kann einem hier öfters einen Streich spielen: So versucht beispielsweise die Workbench bei jedem Öffnen eines Fensters 'irgendwas' auf die Diskette zu schreiben. Was es ist, ist bislang nicht bekannt, wahrscheinlich jedoch Daten über die jeweiligen Fenstergrößen oder ähnliches. Klappt dieses Schreiben jedoch nicht, scheint es auch nicht schlimm zu sein. (Wer näheres darüber weiß, bitte melden.)

Der Pest der Bootblock-Viren kann man mit dieser Schaltung jedoch auf jeden Fall Herr werden, wenn man von Zeit zu Zeit mal eine Probedisk einlegt und auf die Reaktionen wartet.

Klein und fein: Die Schaltung

Zur Erkennung der Zugriffe werden die Signale /DKWEB und /TK0 an den Pins 15 und 16 des Floppyports ausgewertet, die Stromversorgung erfolgt über die Pins 12 (+5 V) und 3 (Masse) des Ports. Die Leitung /DKWEB ist das Write Gate; solange es auf 0 liegt,

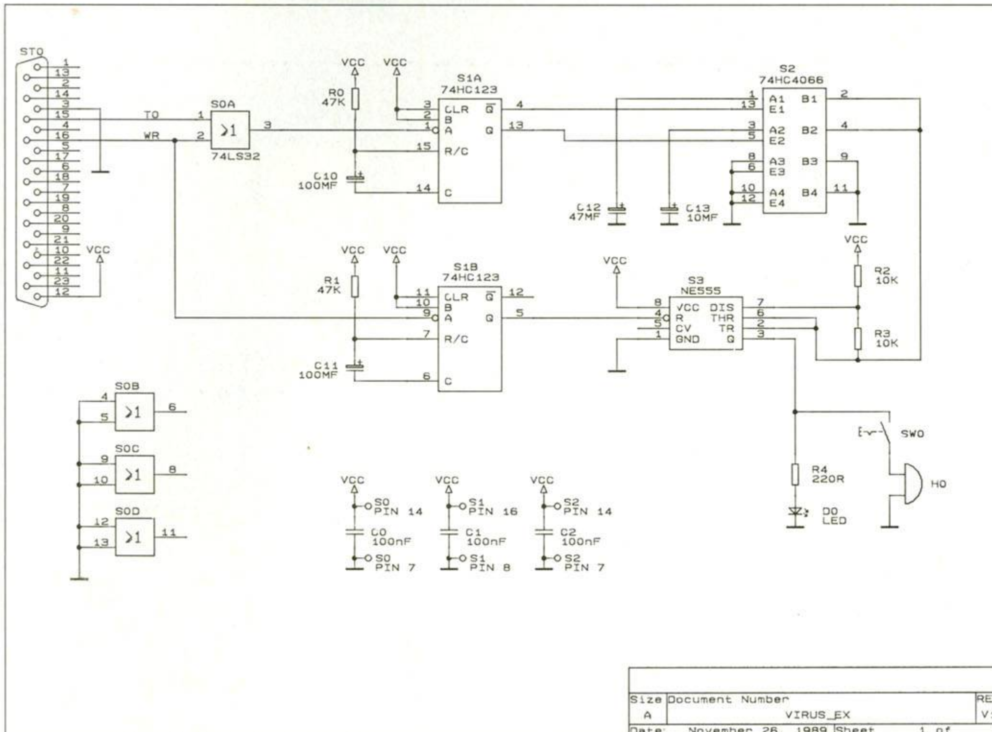


Bild 1: Der Schaltplan des Viren-Warners – drei TTL-ICs und etwas drumherum

werden Daten geschrieben. Über /TK0 wird signalisiert, daß sich der Schreib/Lesekopf des (eines) Laufwerkes gerade auf Track 0 befindet. Geht /DKWEB auf 0, wird über die monostabile Kippstufe S1B das Signal auf mindestens zwei Sekunden verlängert und dadurch der für das periodische Blinken und Piepsen zuständige Multivibrator NE 555 (S3) freigegeben. Wird neben dem Schreiben auch noch das Erreichen von Track 0 signalisiert (Verknüpfung durch S0A), wird das Monoflop S2A aktiviert, welches neben der Freigabe des NE 555 auch dessen Frequenz über S2 und C12/C13 ändert. Schalter SW0 dient zur Abschaltung des Piezosummers H0 und zur Einhaltung der Nachtruhe...

Der Aufbau ist unkritisch und kann problemlos auf einer Lochrasterplatte erfolgen. Der Anschluß erfolgt über ein 20 bis 50 cm langes Kabel und einen 23-poligen Sub-D-Stecker an den Floppyport. Die Bauteile sind frei im Handel erhältlich; bei den TTL-ICs sind LS-, besser noch HC-Typen zu empfehlen, eine Low Current-LED (besondere LEDs mit niedrigem Stromverbrauch) ist auch nicht zu verachten. Zu erwähnen ist noch, daß weder Autor noch Redaktion irgendwelche Garantien übernehmen können, Bau und Betrieb erfolgen also auf eigenes Risiko. Dieses läßt sich durch sorgfältiges Arbeiten stark reduzieren, vielleicht testet man das Elaborat auch erst mal ohne Rechner (fehlerhafte Aufbauten können einiges zerstören – man sollte sich nicht zu sehr auf undokumentierte Sicherungen verlassen, wie wir sie schon kennen... (siehe AMIGA DOS 2/90)).

Für Gelegenheitsbastler...

Die Arbeit mit Fädelstift und Lochrasterplatte ist sicherlich nicht jedermanns Sache. Für diese Fälle wurde das Layout einer kleinen Platine erstellt, mit der die Schaltung sauber und stabil aufgebaut werden kann. Da das Layout recht kompakt ist, sollte zum einen beim Löten sorgfältigst gearbeitet werden, zum anderen müssen die in der Teileliste angegebenen Rastermaße der Bauteile eingehalten werden. Ist die Platine fertig, sollte sie zunächst ohne Rechner getestet werden. Dazu werden die fünf Volt Versorgungsspannung aus einem Netzteil an den Anschlüssen VCC (+) und GND (Masse) angelegt. anschließend wird Anschluß WR auf Masse gelegt. Die LED sollte nun blinken und der Piezo im gleichen Takt piepsen. Wird nun auch T0 auf Masse gelegt, sollte sich dieser Takt erhöhen, wenn WR wieder ohne Massepegel ist, sollte nach etwas Nachlaufzeit wieder Ruhe herrschen. Wenn es nur blinkt, ist entweder der Piezo defekt oder der Schalter in AUS-Stellung. Fehlt jede optische Meldung, so ist die LED wahrscheinlich falsch gepolt. Verlief der Test zur Zufriedenheit und lebt das Netzgerät noch (mehr als 50 bis 150 mA sollte die Schaltung nicht ziehen, je nach Art der Bausteine), kann das Verbindungskabel gelötet werden und der Einbau in ein kleines Gehäuse erfolgen. Die Länge des Kabels sollte dabei 50 cm nicht überschreiten, da es sonst zu Störungen kommen könnte. Dann ist der große Augenblick da: Bei abgeschaltetem Rechner die Schaltung an den Floppyport anstecken, Rechner einschalten und hochfahren und zum ersten Test eine Diskette formatieren. Der Indikator sollte zunächst schnell piepsen und dann beim Wechsel auf Track 1 langsamer. Klappt dies, kann der Lieblingsvirus geladen und probehalber auf einige leere Disketten losgelassen werden...

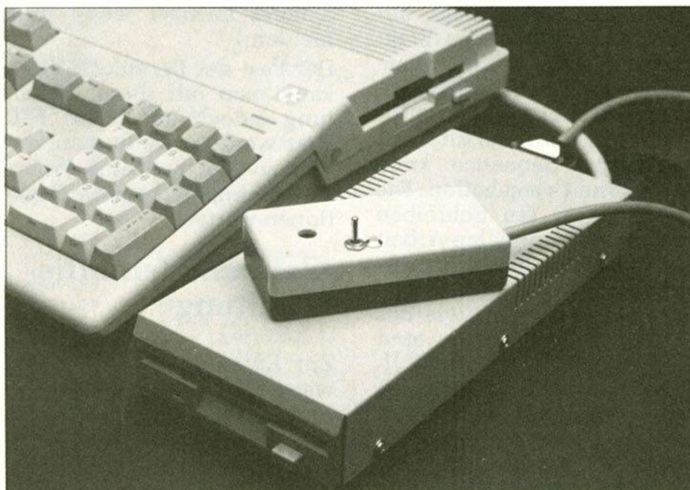


Bild 2: Die Schaltung läßt sich leicht auf einer Lochrasterplatte aufbauen und in einem kleinen Gehäuse in Sicht- und Hörweite unterbringen

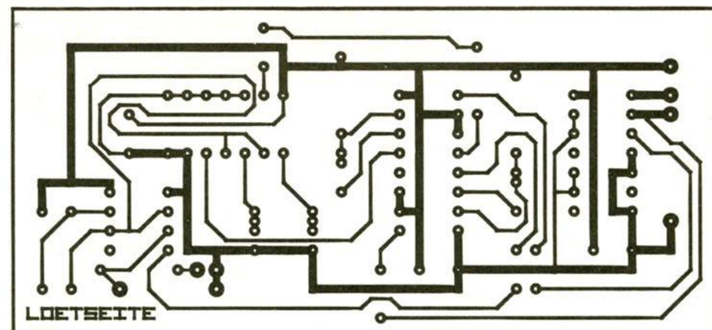


Bild 3: Wer es gerne edel mag, kann statt einer Lochrasterplatte auch eine Platine ätzen. Hier sind Layout...

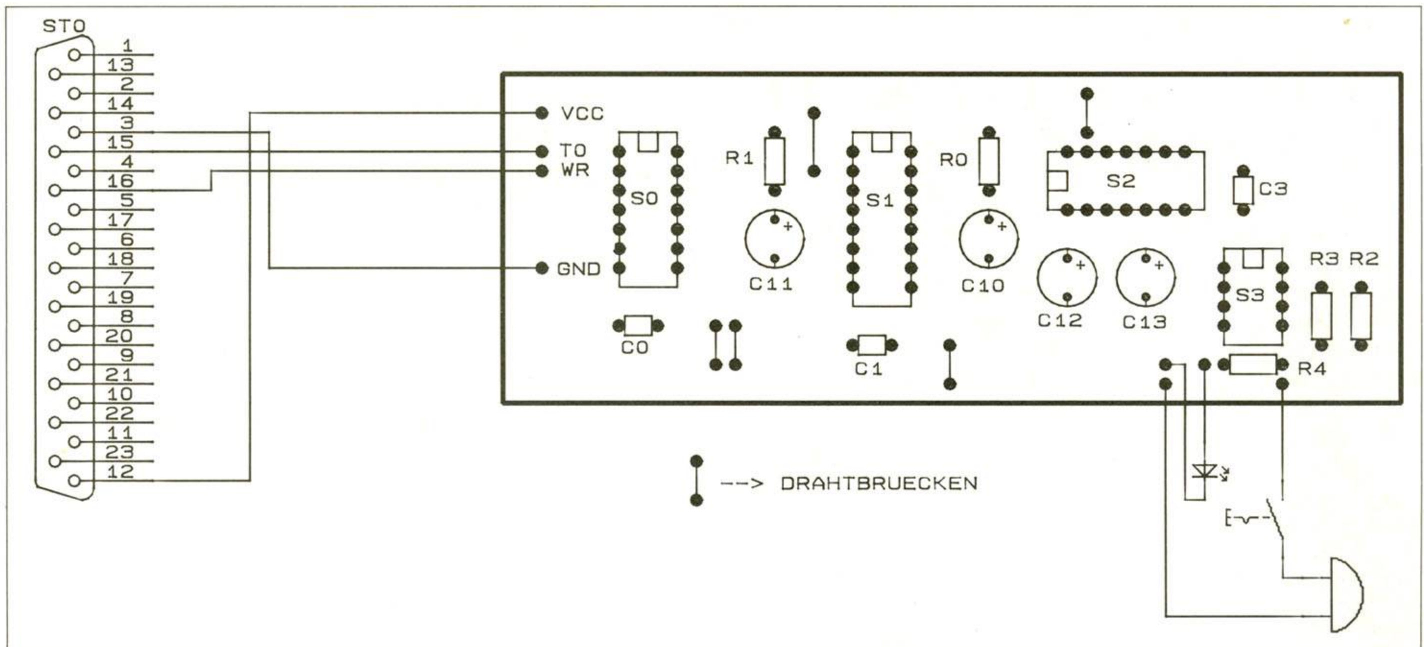


Bild 4:...und Bestückungsplan

Wer schon mit dem Lötkolben in der Hand geboren wurde, wird sich wohl gleich an die Arbeit machen können, wahrscheinlich wird er auch auf eine ordentliche Platine verzichten. Für all jene, die noch nicht so recht wissen, an welcher Seite man den Lötkolben anfaßt, hier noch einige Tips. Wer sich noch nie die Finger verbrannt hat, sollte lieber auf einen Bekannten mit entsprechenden Narben um Hilfe bitten – es ist im eigenen Interesse! Ansonsten gilt die Regel: Von unten nach oben gehen, die niedrigen Bauteile also zuerst einlöten. Das wären in diesem Fall die fünf Drahtbrücken, als nächstes kommen die Widerstände und Fassungen für die ICs an die Reihe. Danach die Kondensatoren und die Elkos, bei letzteren ist auf die Polarität zu achten! Für die Verbindungen zur Außenwelt setzt man entweder kleine Lötnägel ein, oder macht aus dünnem Draht kleine Schlaufen. Nach der optischen Kontrolle auf schlechte Lötstellen oder Kurzschlüsse werden die LED (die Polarität ist auch hier zu beachten) sowie der Piezo mit Schalter durch kurze Verbindungskabel mit der Platine verbunden. Nun werden die ICs eingesetzt, wobei zu beachten ist, daß jedes an die richtige Stelle kommt, ordentlich sitzt, und daß die Kerbe auf der Oberseite des IC in die Richtung zeigt, die auf dem Bestückungsplan angegeben ist. Nun sollte der oben erwähnte Trockentest stattfinden. Ist er absolviert, kann das Verbindungskabel gelötet, dessen Stecker mit einem Gehäuse versehen und das andere Ende mit der Schaltung verbunden werden. Die Schaltung sollte jetzt noch ein kleines Gehäuse bekommen – je nach Geschmack des Anwenders.

Und das wär's auch schon. Ab jetzt herrscht Glasnost im Amiga, denn kein Virus kann sich mehr unauffällig verbreiten, zumindest nicht auf Disketten. Zum Schutz von Festplatten kann dieses Prinzip wohl ebenfalls verwendet werden, da hier ähnliche Leitungen vorhanden sind. Dies bedarf jedoch noch einer genaueren Prüfung.

(jb)

Stückliste VIRUS__EX V1
Stand: 26.11.89

Referenz	Bezeichnung	Rastermaß
C0,C1,C2	Kondensator 100 nF / 63V	5 mm
C10,C11	ELKO 100 µF / 16V	3.5 mm
C12	ELKO 47 µF / 16V	3.5 mm
C13	ELKO 10 µF / 16V	2 mm
D0	Leuchtdiode	2.5 mm
FS0,FS2	IC- Fassung 14 polig	2.5 mm
FS1	IC- Fassung 16 polig	2.5 mm
FS3	IC- Fassung 8 polig	2.5 mm
H0	Piezo- Signalgeber	
R0,R1	Widerstand 47 kΩ 1/4 Watt	10 mm
R2,R3	Widerstand 10 kΩ 1/4 Watt	10 mm
R4	Widerstand 220 Ω 1/4 Watt	10 mm
S0	4-fach OR-Gatter 74 LS 32	2.5 mm
S1	2 Retriggerbare Monoflops 74HC123	2.5 mm
S2	4-fach Bilateral Schalter 74HC4066	2.5 mm
S3	Präzisions-Timer NE 555	2.5 mm
SW0	Schalter (einpolig)	
ST0	SUB-D Stecker 23-polig	
	SUB-D Gehäuse 23-polig	
	vieradriges Kabel (20-50 cm) und	
	Gehäuse nach Belieben	

Sollten HC-Bausteine nicht zur Verfügung stehen, können auch LS-Bausteine benutzt werden:

74 HC 123 – > 74 LS 123 / 74 HC 4066 – > C-MOS Typ CD 4066

Bei den Widerständen handelt es sich um Metallschichtwiderstände.
Der Vorschlag für die LED ist unverbindlich, in diesem Fall wurde eine rote LED mit Chromfassung genommen.
Die Bauteile sind in jedem gut sortierten Elektronikladen oder -versand zu bekommen und kosten zirka 30 DM.

Bild 5: Die Liste der nötigen Bauteile

Edgar Meyzis

Aufsteigerwissen Assembler

Für den dritten Teil des Assemblerkurses steht die Ausstattung der Paint Box mit Tools im Mittelpunkt, um mit Systemroutinen aus "Graphics" und aus "Intuition" vertraut zu werden. Wir bauen einen eigenen Regenbogen und schaffen uns verschieden breite Pinsel. Weiterhin liefern wir Erläuterungen zur Menüauswertung der letzten Folge nach.

Kursfahrplan

Teil 1. Einführung und Grundlagen
Teil 2. Vertiefung der Programmiertechnik
Teil 3. Praktische Arbeit mit Graphics
Teil 4. Praktische Arbeit mit DOS
Teil 5. Hardwarenahe Programmierung von Grafik

Hoffentlich ist es Ihnen nach dem Durcharbeiten der letzten Folge wirklich gelungen, die Wüste zu begrünen, oder liefen etwa die "Kamele" nicht zum Stall, sondern im "Zickzack" in die Arme von Gurus? Falls es Sie interessiert, in Tabelle 1 können Sie die Antworten auf die Fragen finden, die wir zum Abschluß der letzten Folge stellten. Wir haben Sie mit zehn Listings stark strapaziert, und noch immer ist das Ende der Wüstendurchquerung nicht zu sehen. Aber warten wir es ab, heute kommt es noch bunter.

Menüauswertung

Vorher wollen wir aber die Menüauswertung durchsprechen, um sicherzustellen, daß Sie für die weitere Programmierung von Menüs gut gerüstet sind.

Über die Anwahl von Menüs informiert "Intuition" (eine Task des Betriebssystems, der die Priorität 50 zugeordnet ist) ein Anwenderprogramm, wenn das "IDCMP-Flag" mit der Bezeichnung "MenuePick" in der Struktur gesetzt ist, die das aktuelle Window beschreibt. Wir besorgten das in unserem Hauptmodul "P2" in der Routine "WindowAnlegen". bei Wahl eines Menüs sendet Intuition eine "Message" (Nachricht) der Klasse "MenuePick" an das Fenster.

In der Routine "ProgramControl" (P2) beginnt die Auswertung der Message nach der Anweisung "CallExec ReplyMsg". Wenn die erhaltene Nachricht der Klasse "MenuePick" entspricht, dann wird der Inhalt der Message, der Code, ausgewertet. Der Code gibt Aufschluß über das angewählte Menü, den Menüpunkt (MenuItem) und gegebenenfalls dem Menüunterpunkt (SubItem). Es sind somit drei Informationen im Code enthalten, aufgeteilt auf 16 Bits. Die Bits null bis vier enthalten die Nummer des Menüs, die Bits fünf bis zehn die Nummer des Menüpunktes und die restlichen Bits die Nummer des Menüunterpunktes.

Über die Extrahierung der Informationen aus dem Code der Message ist schon viel geschrieben worden. Wir verzichten auf Tricks und machen es ganz einfach, damit unser Programm auch noch nach einem Jahr für uns verständlich bleibt. Zunächst kopieren wir den Code (der sich bereits in Register d6 (MsgCode) befindet) auf den Stack. Dann löschen wir mit **"and.w #$1F,MenuNr"** die Bits 5 bis 15, um die Menünummer zu erhalten. Anschließend laden wir den Code vom Stack in das Register d5 (ItemNr), schieben mit "asr.w #5,ItemNr" die "Menü-Bits" heraus und löschen mit "and.w #$3F,ItemNr" die Bits 6 bis 15. Und wie würden Sie die Nummer des Menüunterpunktes extrahieren? Einen einfachen Weg dazu finden Sie in Tabelle 2, Pkt. a.

Tool Nr. 1, "FreeDraw"

Das erste Werkzeug der Paint Box haben wir Ihnen bereits in der letzten Folge mit der Routine "FreeDraw" vorgestellt. Dazu sind trotz üppiger Kommentierung des Listings einige Informationen nachzutragen: Anhand der Routine "FreeDraw" haben wir das Anlegen "lokaler" Variablen auf dem Stack mittels "link" und "unlk" (siehe Tabelle 1) kennengelernt. Bei Eintritt in die Routine wird zuerst die Zeichenfarbe "Grün" gesetzt. An dieser Stelle werden wir später ansetzen, denn eine "grüne" Wüste wäre auf Dauer langweilig. Ein Farbtopf muß her.

Im nächsten Schritt ändern wir die "IDCMP-Flags", da wir zum Zeichnen Informationen über die Maus benötigen, möglichst immer dann, wenn sie bewegt wurde. "DeltaMove" und "MouseMove" sind die gewünschten Nachrichten.

Kaum sind die Nachrichtenwünsche geäußert, schon geht die Routine in den Wartezustand über und lauert auf den ersten Tastendruck durch Verzweigung in die Routine "LeftButtonPressed". Nach Erhalt der Nachricht über den ersten Tastendruck wird in die eigentliche "Zeichenschleife" eingetreten, deren Anfang mit der Marke "\1" markiert ist. Bevor der erste Bildpunkt gesetzt wird, erfolgt nochmals eine Abfrage, ob der Mausknopf etwa schon wieder gedrückt wurde. Sie mögen einwenden, daß dieses überflüssig sei, weil man in der Wüste ohnehin stets freie Fahrt hat. Stimmt schon, aber irgendwie müssen wir aus der Zeichenschleife wieder herauskommen. Dazu wurde die Nachricht über einen weiteren Druck auf den linken Mausknopf bestimmt.

Die Abfrage des Window-Port kennen wir bereits aus der Routine "ProgramControl".

Es wäre ohne weiteres möglich, sämtliche Abfragen des Port in einer zentralen Routine vorzunehmen. Wir haben darauf verzichtet, um die Routinen weitestgehend zu entkoppeln und somit überschaubar zu halten.

Nun wird es endlich grün. Die aktuellen Positionen des Mauszeigers sind dem Window zu entnehmen. Sicherlich wird es jetzt auch verständlicher, warum wir in der letzten Folge die Window-Struktur definiert haben.

Sollte es Ihnen nicht gelingen, ruckfrei zu zeichnen, dann schalten Sie bitte Ihren Mausbeschleuniger aus. Einige praktische, jedoch kaum spürbare Verbesserungen sind durch Straffung des Assemblercodes möglich. Was würden Sie tun? Anregungen finden Sie in Tabelle 2, Pkt. b.

Angemerkt sei auch, daß die Systemroutine "WritePixel" auf unsere Zeichenaufgabe hin optimiert werden könnte. Die Routine ist universell ausgelegt, um jeder Bildschirmgröße und Kombination von Bildebenen zu genügen. Wir haben eine Spezialanwendung, könnten somit an Geschwindigkeit durch eine spezialisierte, eigene hardwarenahe Routine gewinnen. In der letzten Folge gehen wir darauf ein. Vergessen wir nicht, unser Ziel ist es, möglichst sicher anzukommen. Assembler ist schnell genug.

Nun wird es bunt

Zu "FreeDraw" bleibt noch auszuführen, daß entsprechend unseres Schichtenmodells, dem wir ja schon häufig begegnet sind, vor Verlassen der Routine die "IDCMP-Flags" wieder in den Zustand zu versetzen sind, der bei Eintritt bestand. Dieser Beitrag zur sicheren und übersichtlichen Programmierung kostet mitunter seinen Preis. Es könnte z.B. der Fall eintreten, daß die "IDCMP-Flags" unnötig häufig verändert werden. Nun könnte man ja den Port durch Setzen der entsprechenden Flags für sämtliche Nachrichtenklassen "empfänglich" machen, die in dem Programm vorkommen. Damit wäre der Nachteil verbunden, daß an den Port unnötige Messages gesandt werden, die unser Programm alle analysieren müßte, um nichts zu verpassen. Vorteilhafter ist es, den Nachrichtenstrom so zu steuern, daß möglichst nur benötigte Informationen darin enthalten sind.

“Grün“ ist durchaus in und regt auch an bzw. auf, jedoch nur, weil es noch andere Farben gibt. Die wollen wir uns jetzt verschaffen, aber nicht, indem wir die Ölwanne auf einem Stein bersten lassen und uns an dem schillernden, “staubfreien“ Teil der Wüste erfreuen. Wie denn? Erinnern Sie sich noch an die erste Folge, in der wir feststellten, daß die vier Bildebenen unseres Screen es ermöglichen, 16 Farben darzustellen. Mit Hilfe eines Menüs wollen wir uns den Zugang zu diesen Farben verschaffen, um sie z.B. in unserer Zeichenroutine einzusetzen. Dazu benennen wir das zweite Menü in “Farben“ um und versehen es mit 16 Menüpunkten. Die ordnen wir aber nicht senkrecht untereinander an, sondern in zwei Zeilen zu jeweils acht Menüpunkten. Die aktuelle Farbe gilt es zu kennzeichnen. Dazu setzen wir die bekannten “Menü-Häkchen“ ein. Es wäre auch möglich gewesen, ein anderes, selbst angelegtes Image zu verwenden, z.B. einen Rahmen. Die aktuelle Farbe legen wir in einer globalen Variable ab und nutzen den enthaltenen Farbwert künftig bei der Arbeit mit der Paint Box. Ach ja, und dann ist da noch das “mutualExclude“ (einvernehmliches Ausschließen), eine spitzenmäßige Dienstleistung von “Intuition“, die bei Auswahl einer Farbe messerscharf schließt, daß die vorhergehende Farbe nicht mehr aktuell ist, und das “Menü-Häkchen“ entfernt. Wir müssen “Intuition“ allerdings auf diese Aufgabe vorbereiten.

Nach dem Überblick wenden wir uns nun der Programmierarbeit zu. In der Datei “Intuition.i“ sind die “Menu Item Flags“ um die Definition “checked = $100“ zu ergänzen. Das Flag wird benötigt, um den Ausgangszustand des Farbmenüs zu initialisieren (Farbe null “abgehakt“). Welche Farbe befindet sich eigentlich im Farbregister null? Keine Ahnung? Na, dann schauen Sie sich doch bitte noch einmal in “P2“ die Routine “ScreenAnlegen“ an.

Im nächsten Schritt nehmen wir uns das Hauptprogramm vor, das ab sofort mit “P3“ angesprochen wird. Zunächst ändern wir den “ScreenTitle“ in “Paint Box 0.3“ um. Dann ergänzen wir die Struktur der globalen Variablen um das Element “word FarbNr“, das wir als eigenständige Zeile vor “struktLaenge globals__size“ einfügen. Auf die aktuelle Farbe können wir künftig mit “FarbNr(globals)“ zugreifen. Bislang gibt es nur einen Bedarfträger dafür, die Routine “FreeDraw“. Ersetzen Sie bitte darin die Anweisung “move.w #14,d0“ durch ? Ja was denn? Die Antwort enthält Tabelle 2, Pkt.c. Und wie kommt der Farbwert in die globale Variable? Die Aufgabe übernimmt die Routine “Menu1Auswerten“. Es ist lediglich die erste Zeile durch folgende Anweisung zu ersetzen “move.w 4(sp), FarbNr(globals)“. Erinnern Sie sich! Wir wollen 16 Menüpunkte anlegen, die von 0 bis 15 durchnumeriert sind. Die Nummer des angewählten Menüpunktes ist somit zugleich die Nummer eines Farbregisters. So einfach ist es, den Farbwert aus dem Menü zu gewinnen.

Da wir gerade bei “P3“ sind, wollen wir die Lesbarkeit der Routine “ProgramControl“ noch leicht verbessern. Ersetzen Sie bitte die Zeile “btst #0,MenuNr ...“ durch “cmp.w #1,MenuNr ...“. “P3“ sollte sich nun wie gewohnt assemblieren lassen. Um “P3“ zu testen, könnten Sie der globalen Variable “FarbNr(globals)“ vor Eintritt in die Routine “ProgramControl“ einen Wert zuweisen. Verzichten Sie beim Test aber darauf, das Menü “Zeichnen“ anzuwählen.

Als nächstes sollten Sie sich die Datei “M2“ vornehmen und durch Ergänzung gemäß Listing 1 (L1) zu “M3“ umbauen. Zuerst taufen wir das zweite Menü in “Farben“ um und legen zugleich “FarbText“ global an (drei Leerzeichen, siehe die letzten vier Zeilen von L1).

Wie Sie erkennen, haben wir die Routine “MenuAnlegen“ umgebaut. Wir steuern künftig von hier aus nur noch das Anlegen der einzelnen Menügruppen, d.h. “ProjektMenuAnlegen“ und “FarbMenuAnlegen“. Die Routine “ProjektMenuAnlegen“ enthält die überwiegenden Teile der Routine “MenuAnlegen“ der vorherigen Version.

Neu ist die Routine “FarbMenuAnlegen“, die ihre ersten Zeilen gleichfalls aus der früheren Version von “MenuAnlegen“ bezieht. Des weiteren übernimmt sie es, das Anlegen des ersten SubItems zu steuern sowie 15 weiterer in einer Schleife. Die eingesetzten Routinen sind hinreichend kommentiert. Lediglich auf das “mutualExclude“ soll noch eingegangen werden. Diese Variable ist 32 Bits breit und dient “Intuition“ zu erkennen, in welchen anderen SubItems die Flag “checked“ und damit das “Häkchen“ gelöscht werden soll. Es können somit 32 SubItems verwaltet werden. Ein Häkchen kann gesetzt werden, wenn das zum SubItem korrespondierende Bit gelöscht ist. Wir haben 16 Menüpunkte und initialisieren die Variable in “ErstesFarbItemInit“ mit $FFFE (Exclude-Maske). Von den unteren 16 Bits, die uns hier nur interessieren, ist somit das Bit 0 ungesetzt. Beim Anlegen weiterer Menüpunkte (“WeiteresFarbItemInit“) setzen wir die Anweisung “rol.w mi__mutualExclude + 2(item)“ ein, um die “Exclude-Maske“ im Uhrzeigersinn zu rotieren. Warum eigentlich “mutualExclude + 2“? Tabelle 2, Pkt. d, liefert die Antwort.

Ein eigener Regenbogen

Die Integration von “M3“ in unsere Paint Box sollte keine Probleme bereiten. Falls ja, dann empfehlen wir, schrittweise vorzugehen. Setzen Sie zunächst nur die Routine “ErstesFarbItemInit“ ein. Wenn das o.k. ist, dann sollten Sie “WeiteresFarbItemInit“ aktivieren, ohne in die Schleife zu gehen. Anschließend aktivieren Sie die Schleife, um die Paint Box um eine wesentliche Eigenschaft zu bereichern.

1. Die Parameterübergabe erfolgt über den Stack. Mit der Anweisung “move.w ItemNr,-(sp)“ wird die Nummer des “MenuItem“ auf dem Stack abgelegt und mit “move.w 4(sp),d0“ in den Routinen “Menu0Auswerten“ bzw. “Menu1Auswerten“ verfügbar gemacht.

2. Stack:

Stack	
	« – sp vor Ablegen MenuItem
ItemNr	
	« – sp + 4
Rücksp.	
	« – sp bei Eintritt in Routine

Mit der Anweisung “move.l (sp) + ,a0“ wird die Rücksprungadresse vom Stapel entfernt. Die Adresse gelangt mit der Anweisung “bsr ...“ auf den Stack. Der Rücksprung führt zur Fortsetzung des Programms mit der Anweisung, die unmittelbar dem “bsr ...“ folgt. Vor dem Rücksprung wird noch der Stack-Pointer (sp) mit der Anweisung “addq.l #2,sp“ um den Platz berichtigt, den der Parameter “ItemNr“ belegte. Nach Abarbeitung der Routine weist der sp wieder exakt auf dieselbe Speicheradresse wie vor dem Ablegen des Parameters.

Alternativ zum gewählten Vorgehen hätte auch der Rücksprung mit “rts“ und anschließender Berichtigung des Stack (“addq.l #2,sp“) erfolgen können. Beide Verfahren sind bei der Implementierung von Hochsprachen gebräuchlich. Das erste Verfahren benötigt 28 Taktzyklen, das zweite hingegen nur 24.

3. Die Anweisung “link a2,#lb__size“ legt den Inhalt des Registers a2 auf dem Stack ab. Anschließend wird der Wert des sp in a2 übernommen, danach sp um sechs verringert. So wird auf dem Stack Platz für die Aufnahme der lokalen Variablen “lb__inMsg“ (vier Bytes) und “lb__pressed“ (ein Byte) geschaffen. Um die Ausrichtung des Stack auf gerade Adressen zu erhalten, wurden insgesamt sechs Bytes reserviert.

Die Offsets sind negativ, weil a2 auf die Basis des “frame“ (engl. Bezeichnung für den reservierten Bereich) weist und der Stack nach unten wächst (Konvention).

Das Register a2 wird als Basisregister für die Adressierung der lokalen Variablen eingesetzt.

Die Anweisung “unlk a2“ bewirkt, daß das Register den Wert erhält, den es vor Ausführung von “link ...“ innehatte. Der reservierte Stackbereich wird freigegeben und der sp mit der Adresse versehen, auf die er vor dem Aufruf von “link ...“ wies.

Die Verwendung von “link“ und “unlk“ in der Assemblerprogrammierung ist nicht häufig anzutreffen, obwohl sie weniger Taktzyklen (16 bzw. 12) benötigt als eine entsprechende Nachbildung (move.l ax,-(sp) / addq.l #6,sp / move.l (sp) + ,ax erfordern 12 + 8 + 12 Zyklen).

Tab. 1: Antworten zu den Fragen aus Folge 2

Wer besitzt schon einen eigenen Regenbogen? Wir bieten Ihnen die Möglichkeit, sich die Farben selbst zusammenzustellen, mit denen die Paint Box arbeiten soll. Wie würden Sie vorgehen? Hilft ein Blick auf den Text der Routine "ScreenAnlegen" (P3) weiter? Na klar! Wir haben lediglich die 16 Farbregister, mit denen unser Screen arbeitet, mit Farbwerten nach unserem Geschmack zu versorgen. Genau das leistet die Routine "FarbRegisterInit" (Listing 2) für uns. Sie liest aus der Tabelle "NeueFarben" jeweils die Nummer des zu ändernden Farbregisters und die Farbanteile Rot, Grün sowie Blau in die Prozessorregister d0 bis d3, um anschließend mit "SetRGB4" die gewünschte Änderung herbeizuführen. Ganz schön rasant, die Ladung, nicht die unseres Wüstenautos, sondern die der CPU-Register (vier auf einen Streich). Die Farbwerte der Tabelle "NeueFarben" entstammen der neuesten Auflage von [1]. Die Farbregister können wir beliebig ändern, weil uns ein eigener Screen zur Verfügung steht. Der "Workbench Screen", unsere Startplattform, bleibt unangetastet.

Sollten Sie die vorgestellte Routine in Ihrer Paint Box verwenden wollen, dann fügen Sie sie bitte nach "WindowSchliess" ein. Die Tabelle "NeueFarben" sollte ihren Platz unmittelbar vor dem "end" von P3 finden. Die sieben Codezeilen der Routine "ScreenAnlegen", mit denen wir den Hintergrund schwärzten, benötigen Sie nicht mehr. Rufen Sie statt dessen "FarbRegisterInit" auf.

Der Regenbogenmacher läßt sich optimieren. Wie würden Sie vorgehen? Tabelle 2, Pkt. e, enthält einen Vorschlag dazu.

Gemeinsamkeiten erkennen und nutzen

Auch wenn man sich auf einer Wüstenfahrt gegenseitig helfen muß, soll jetzt nicht auf ein soziales Thema ausgewichen werden. Es geht vielmehr darum, ähnliche Aufgabenstellungen in einem Programm zu erkennen und allgemein nutzbare Routinen daraus zu entwickeln. Programme werden kürzer und auch sicherer, wenn auf ausgetestete Teile zurückgegriffen werden kann. In einem nächsten Schritt wollen wir es der Paint Box beibringen, Linien zu ziehen. Es liegt nahe zu prüfen, ob die bereits bewährte Routine "FreeDraw" Teile aufweist, die in jeder Zeichenroutine benötigt werden. Ein Blick auf den Quelltext zeigt, daß mit Sicherheit das Setzen des Farbstiftes, die Wahl des Zeichenmodus und die Anforderung von Meldungen über Mausaktivitäten (Wüstenmäuschen oder wüste Mäuschen?) von allgemeinem Interesse sind. Auch die "Abbestellung" von Mausinformationen am Schluß von "FreeDraw" gehört dazu. Wir ziehen aus "FreeDraw" die allgemein interessierenden Teile heraus (siehe Listing 3) und rufen die neu entstandenen Routinen "ZeichnenVorbereiten" sowie "ZeichnenAbschliessen" statt dessen auf.

Immer schön geradeaus...

... kann man selbst durch die Wüste nicht fahren. Manchmal ist es sogar wünschenswert, mehrere Routen zu planen. Genau das leistet die nun vorzustellende Routine "ZeichneLinie" aus Listing 4. Die gewählte Vorgehensweise bietet eine gute Basis für die Entwicklung weiterer Zeichenroutinen (Vierecke, Kreise, Ellipsen, Vielecke). Der Code ist von den beiden gerade vorher besprochenen Routinen eingerahmt.

Ein Druck auf den linken Mausknopf erweckt die Routine zum Leben. Sie markiert die aktuelle Mausposition mit einem Punkt in der Farbe der zu zeichnenden Linie (WritePixel) und stellt den Zeichenmodus auf invers (SetDrMd). Weitere Zeichenaktivitäten erfolgen nun nicht in einer bestimmten Farbe. An den betroffenen Punkten wird die Bildschirmfarbe einfach durch Tausch der Wertigkeit der Bits in den Bitplanes invertiert. Nach doppeltem Invertieren leuchtet die ursprüngliche Farbe wieder. Diesen Effekt machen wir uns zunutze. Der interne Cursor ("Graphics") wird auf den farbig markierten Ausgangspunkt genagelt ("Move"), die veränderte Mausposition als Endkoordinate gesetzt und eine invertierte Linie gezogen ("Draw"). Ein zweiter Linienzug mit denselben Koordinaten löscht die Linie wieder. Diese Technik ermöglicht es uns, eine vorläufige Linie zwischen dem Ausgangspunkt und der jeweiligen Mausposition wie ein Gummiband zu spannen. Erst wenn der linke Mausknopf ein zweites Mal gedrückt wird, erscheint die Linie in der gewünschten Farbe. Der Spuk ist vorbei. Der inverse Zeichenmodus wurde aufgeben.

Etwas Übung haben wir ja schon im "Zickzack" durch die Wüste. Leider liefert beim Linienziehen eine "Zickzacktechnik" keine befriedigenden Ergebnisse. Die invertierten Bildpunkte lassen sich nur sauber zurücksetzen, wenn der zweite Linienzug genau in derselben Richtung erfolgt wie der erste. Ein wenig umständlich, aber zwingend notwendig. Der Grafikcursor ist somit jedesmal auf den Anfangspunkt zu setzen. Es gibt kein Zurück. (Das gilt auch für unseren Kurs. Mehr als die Hälfte der Wüste ist durchquert, wenn Sie nachher das Heft aus der Hand legen.)

Noch ein Menü

Immerhin haben wir jetzt schon drei Zeichenroutinen, "Löschen", "FreeDraw" und "ZeichneLinie" (Seit wann werden Schwarzmaler mitgezählt?). Sollte uns das nicht ein eigenes Menü wert sein? Vielleicht haben Sie ja schon eins dafür angelegt, wie in der ersten Folge empfohlen. Un-

a. Sie könnten wie folgt vorgehen:

```
...
asr.w #5,ItemNr ; MenuNr herausschieben
move.w ItemNr,-(sp) ; für spätere Auswertung
and.w #$3F,ItemNr ; ItemNr isolieren
move.w (sp) + ,d4 ; d4 für SubItem
asr.w #6,d4 ; ItemNr herausschieben
and.w #$1F,d4 ; SubItem isolieren
...
```

Ein einfacher und übersichtlicher Weg. Mit "rol"-Operationen könnten Sie sogar Zeit einsparen. Die Auswertung von Menüs gehört jedoch nicht, laufzeitmäßig gesehen, zu den kritischen Teilen eines Programms.

b. Der Code von "LeftButtonPressed" ließe sich wie folgt straffen:
- Auf "link" und "unlk" verzichten
- "lb__pressed" durch Registervariable ersetzen, z.B. d5
- d5 und d6 mit "movem" bei Eintritt in die Routine auf den Stack retten und später restaurieren

Auch "FreeDraw" kann im Ablauf beschleunigt werden, indem man die Register der CPU stärker einsetzt. Die Register a2 und a3 könnten z.B. die Adressen der Speicherstellen aufnehmen, in denen die Mauskoordinaten von Intuition abgelegt werden.

c. "move.w FarbNr(globals),d0"

d. "mi__mutualExclude" enthält den Offset zu einem Langwort. Wir rotieren jedoch nur die untere Hälfte des Langwortes und müssen deshalb den Offset um zwei erhöhen.

e. In der Schleife von "FarbRegisterInit" wird bei jedem Durchlauf die Adresse des "ViewPort" neu ermittelt. Der Aufwand ist reduzierbar, indem die Adresse vor Eintritt in die Schleife nur ein einziges Mal festgestellt und in einem Register (z.B. d5) zwischengespeichert wird. In der Schleife wird dann am Anfang jeweils der Inhalt von d5 nach a0 kopiert. Die Passage sieht wie folgt aus:

```
    move.l Screen(globals),a0
    lea.l sc__viewPort(a0),a0
    move.l a0,d5
\farbLoop
    move.l d5,a0
    ...
```

Bei dem gewählten Weg sollte auch der Inhalt des Registers d5 bei Eintritt in die Routine auf dem Stack gesichert und bei Verlassen wieder hergestellt werden.

Tab. 2: Wie immer, Antworten auf begleitende Fragen

sere Lösung ist in Listing 5 dargestellt. Nach dem bewährten Muster rufen wir im Modul "M3" in der Routine "MenuAnlegen" einfach "ZeichenMenu auf. Die statischen Menüinhalte fügen wir in "M3" vor "end" ein.

Ein Menü ohne Auswertung ist wie eine Wüste ohne Sand. Listing 6 (einzufügen in P3) stellt ein Vorgehen zur Auswertung des "ZeichnenMenüs" dar. Der Teil, der mit "\M2" beginnt, ist in die Routine "ProgramControl" vor der Marke "MouseBut" einzusetzen. In der zweiten Zeile nach "\M1" ist dann zu "\M2" zu verzweigen (und nicht zu "control0").

Imagepflege

Vielleicht läßt sich das persönliche Image mit einer Wüstendurchquerung "hochpowern". Darüber soll hier nicht reflektiert werden. Wir benötigen Pinsel, um unser Gepäck vom Staub zu befreien, daß heißt, Farbe auf den Schirm zu bringen. Je breiter der Pinsel, desto einfacher und schneller ist die Arbeit verrichtet. Der Amiga bietet dafür die Image-Technik. Sie ermöglicht es, Bildschirmmuster aus dem Chip-RAM blitzschnell in den aktuellen "Rastport" zu kopieren. Eine ausführliche Beschreibung der Technik findet sich in [3].

Wie kommen die Daten in das Chip-RAM? Unser PD-Assembler "A68k" kennt die Option "CHIP" für die Anweisung "SECTION". Leider ignoriert der Linker den so geäußerten Wunsch nach Chip-RAM. Folglich weisen wir ihn mit der Option "CHIP" explizit an, unser Programm so mit Informationen zu versehen, daß es beim Aufstart in das Chip-RAM geladen wird. Ein typischer Aufruf des Linkers könnte jetzt wie folgt aussehen:

```
BLINK obj/P3.o obj/I3.o obj
/M3.o obj/G3.o TO bin/P3
CHIP
```

In Listing 7 ist ein ausbaufähiger Weg zur "Imagepflege" dargestellt. Der obere Teil enthält mit "DrawImage" und der Definition der Struktur "Image" Informationen, die in die Datei "Intuition.i" einzufügen sind. Der folgende Teil (unterhalb der markanten Trennlinie) ist selbständig assemblierbar.

Um den Code nicht nur als blinden Passagier in unserem Wüstengefährt mitzuführen, ist eine Integration mit den übrigen Programmteilen durchzuführen. Unter den Importen aus dem Hauptprogramm erkennen Sie neu "Pinsel1", "Pinsel2" und "Brush". Dieses sind Offsets innerhalb unseres Verbundes "globals". In "P3" sind die Offsets als "aptr" nach bewährtem Muster vor der Zeile "word FarbNr" einzufügen. Und was ist noch zu tun, um die neuen Errungenschaften von "globals" publik zu machen?

In "I3", so bezeichnen wir das neue Modul, übernimmt die Steuerung aller Aktivitäten die Routine "PinselInit". Sie ist daher mit "PUBLIC" zu exportieren und entsprechend in "P3" zu importieren. Der Aufruf erfolgt am besten im Hauptprogramm vor

"bsr MenuAnlegen"

mit

"bsr PinselInit"

gefolgt von

"beq.s endeW".

In "I3" haben Sie sicherlich schon die bereits aus "M3" bekannte Routine "MemBlockAnfordern" entdeckt. Einer Kopie der Routine hätte es nicht bedurft. Wir unterlaufen damit auch ein wenig die in Teil 5 dargelegte Philosophie. Andererseits gewinnen wir mit der jetzigen Fassung von "I3" an Übersichtlichkeit. Die Kopplung zwischen selbständig assemblierbaren Programmteilen sollte man möglichst einfach halten, idealerweise nur über das Hauptprogramm. Daran haben wir uns gehalten.

Nun zu den beiden "Pinsel-Routinen". Sie beschaffen den Speicher zur Aufnahme der "Image-Verwaltungsstruktur" und initialisieren diese mit einigen Werten. (Zur Erinnerung: Alle nicht angesprochenen Felder enthalten Null, da der Speicher entsprechend angefordert wurde.) In "Pinsel1Init" wird zum Beispiel bestimmt, daß das zu verwaltende Image nur einen Bildpunkt umfaßt, der durch vier Bildebenen sechzehnfach eingefärbt werden kann. Das Image selbst sieht ganz dürftig aus, wie so häufig, wenn man hinter die Fassade schaut. Es trägt die Bezeichnung "Pinsel" und besteht aus vier Worten, deren sämtliche Bits gesetzt sind. Für den ersten Pinsel benötigen wir nur das höchstwertige Bit des ersten Wortes. Der zweite Pinsel ist nicht mehr ganz so fein. Sein Image erstreckt sich über sechs mal vier Bits, die gleichfalls aus den vier Worten von "Pinsel" herausgeschnitten werden. (Auch in der Wüste gibt es Mitesser.)

Pinsel auf dem Tablett serviert

Nur Majestäten erhalten Pinsel auf dem Tablett serviert. Auf unserer Wüstentour empfinden wir uns als solche und leisten uns den Luxus entsprechend Listing 8. Nach demselben Schema, wie vorher für die "Zeichnen-Menüs" beschrieben, ergänzen Sie bitte "M3". In der Routine "PinselMenuInit" arbeiten wir wieder mit den "Exclude Flags", da ja stets nur ein Pinsel zur selben Zeit genutzt werden kann. Die angewandte "Verriegelungstechnik" ist Ihnen ja schon vom Aufbau des "Farben-Menüs" her bekannt. Sie werden auch noch weitere, bereits bekannte Vorgehensweisen entdecken. Sicherlich hätte man hier und da eleganter vorgehen können. Wir streben in diesem Kurs jedoch Sicherheit im Umgang mit Assembler an. Es bietet sich daher an, auf bewährten Code zurückzugreifen.

Auch Listing 9 bietet keine Überraschungen. Nach erprobtem Muster wird das gerade angelegte Menü ausgewertet. Gehen Sie bei der Integration des Codes so vor wie für Listing 5 beschrieben.

Wir könnten nun zwischen zwei Pinseln wählen, wenn es eine Zeichenroutine gäbe, die die unterschiedlichen Pinselbreiten nutzt. Es bietet sich an, die Routine "FreeDraw" entsprechend zu modifizieren. In Listing 10 sehen Sie das Ergebnis. Wir greifen uns den Pinsel

("move.l Brush(globals),a1"),

tauchen ihn in die Farbe

("move.w FarbNr(globals), ig_planePick(a1)")

und betupfen den Bildschirm damit ("CallIntui DrawImage"). Betupfen ist schon richtig. Der Vorgang wird in einer Schleife so lange wiederholt, bis der linke Mausknopf gedrückt wird.

Am Ende unserer heutigen Wüstentour bleibt festzustellen, daß wir zwar unsere Oase erreicht, aber den Fahrplan nicht ganz eingehalten haben. In einer Wüste kommt das halt vor. Eigentlich wollten wir auch die eigenartigen Füllmuster am Berghang "Area" bewundern. Wir holen dieses noch nach.

Anregungen

Bevor Sie sich nun an den Amiga heranmachen, um die neuen Vorstellungen gegen den Willen der Gurus durchzusetzen, wollen wir Ihnen eine kleine Aufgabe mitgeben, zu der wir die Lösung in der nächsten Folge präsentieren werden: Schaffen Sie bitte eine weitere Zeichenroutine, die anstelle von Linien Vierecke zeichnet. Dabei soll ein Eckpunkt beim ersten Druck auf den Mausknopf markiert werden und das Viereck durch eine gedachte Diagonale zur aktuellen Mausposition definiert sein. Beim Bewegen der Maus soll das Viereck stets in der aktuellen Größe und Form dargestellt werden. Bei Ihrer Lösung könnten Sie gut von der Routine "ZeichneLinie" ausgehen.

Eine weitere Möglichkeit, Ihre Paint Box auszubauen, bestünde in der Entwicklung einer Routine, die Kreise zeichnet. Dabei könnte der Mittelpunkt durch den ersten Druck auf den Mausknopf bestimmt sein und die Entfernung zur aktuellen Mausposition als Radius interpretiert werden.

(mm)

Literaturhinweise:

[1] Amiga Hardware Reference Manual, Ausgabe September 1989

[2] Amiga Intern, Band 2

[3] Amiga Intuition Reference Manual

Listings

```
****************************************************************
*                        AMIGA DOS INFO                        *
*  Programmname          : Listing 1                           *
*  Funktion              : 16 Farben fuer Paintbox             *
*  Sprache               : Assembler                           *
*  Besonderheiten        : Mit A68K erstellt                   *
****************************************************************

ProjektMenuAnlegen
*-----------------
    bsr ErstesMenuInit        ; Menu_0 anlegen
    tst.l Menu(globals)
    beq \M
    bsr ErstesMenuItemInit ; Item00
    beq \M
    bsr WeitererTextInit    ; Item01
    beq \M
    bsr WeiteresItemInit
    ItemTextEintragen Item01,Cmnd01
    beq \M
    bsr WeitererTextInit    ; Item02
    beq \M
    bsr WeiteresItemInit
    ItemTextEintragen Item02,Cmnd02
\M rts

ErsterFarbTextInit ; inTuiText anlegen
*-----------------
    move.l #it_size,d0       ; Speicher anfordern und in
    bsr MemBlockAnfordern    ; Liste verwalten
    beq.s \F                 ; kein Speicher erhalten
    move.l d0,oText          ; sichern fuer Kopie/Eintrag
    move.l d0,a1             ; Zeiger auf angelegte Struktur
    clr.b  it_frontPen(a1)   ; Farbregister null
    clr.b  it_backPen(a1)    ; Farbregister null
    move.b #jam2,it_drawMode(a1) ; zweifarbig zeichnen
    lea.l  FarbText,a0       ; drei Leerzeichen enthΣlt
    move.l a0,it_iText(a1)   ; jede Auspraegung der Struktur
\F rts

farbItemFlags = itemText¦highBox¦itemEnabled¦checkIt
itemBreite = 24              ; 24 Bildpkte=Breite 3 Zeich.
itemAbstand = itemBreite+4 ; 4 Bilpunkte fuer Rand

ErstesFarbItemInit ; SubItem anlegen, Ergebnis in item
*-----------------
    bsr ErsterFarbTextInit   ; SubItem Text anlegen
    beq.s \r
    move.l #mi_size,d0       ; Speicher fuer SubItem anford
ern
    bsr MemBlockAnfordern
    beq.s \r
    move.l d0,item              ; fuer ind. Address. + Erge
b.
    move.w #1,mi_topEdge(item) ; in Bezug auf Menue ausric
hten
    move.w #4,mi_leftEdge(item)
    move.w #itemBreite,mi_width(item) ; und dimensioniere
n
    move.w #8,mi_height(item)
    move.w #farbItemFlags,mi_flags(item) ; Eigenschaft. b
est.
    move.l #$FFFE,mi_mutualExclude(item) ; nur ich
    move.l oText,mi_itemFill(item)       ; Text anhaengen
    move.l item,me_firstItem(menu)       ; ans Menue haeng
en
\r rts

WeitererFarbTextInit ; IntuiText anlegen
*--------------------
    bsr WeitererTextInit
    beq.s \T
    move.l oText,a1
    addq.b #1,it_frontPen(a1)
    addq.b #1,it_backPen(a1)   ; Farbregister erhoehen
\T rts

WeiteresFarbItemInit ; MenuItem anlegen
*-------------------
    move.l #mi_size,d0
    bsr MemBlockAnfordern
    beq.s \i
    move.l d0,-(sp)
    Strukturkopieren item,d0,#mi_size ; Vorgaenger
    move.l (sp),mi_nextItem(item) ; an Vorgaeng. haengen
    move.l (sp),item              ; auf aktuelle Strukt.
    add.w  #itemBreite+4,mi_leftEdge(item) ; hor. Pos.
    cmp.w  #itemAbstand*8+4,mi_leftEdge(item) ; neue Zeil
e ?
    bne.s  \w                      ; nein
    move.w #4,mi_leftEdge(item)    ; ja, an den linken R
and
    add.w  #10,mi_topEdge(item)    ; eine Zeile tiefer
\w rol.w   mi_mutualExclude+2(item) ; nur ich, kein ander
er
    move.l oText,mi_itemFill(item)  ; Text anhaengen
    clr.l  mi_nextItem(item)        ; zunaechst kein Nachf
.
    move.l (sp)+,d0
\i rts

FarbMenuAnlegen ; steuert Anlegen der Menuegruppe "Farben
"
*--------------
    bsr WeiteresMenuInit     ; Menu_1 anlegen
    beq.s \a
    MenuBenennen Menu_1
    bsr ErstesFarbItemInit
    beq.s \a
    moveq #14,d2          ; 15 weitere SubItems sind anzuleg
en
\farbItemLoop
    bsr WeitererFarbTextInit
    beq.s \a
    bsr WeiteresFarbItemInit
    dbeq d2,\farbItemLoop ; bis alle 16 SubItems stehen
    beq.s \a
    move.l me_firstItem(menu),item ; Adresse erstes SubIt
em
    ori.w #checked,mi_flags(item)  ; Haekchen setzen
\a rts

MenuAnlegen
*----------
    movem.l d5-d6/a2-a6,-(sp)
    bsr ProjektMenuAnlegen
    beq \M
    bsr FarbMenuAnlegen
    beq \M
    bsr MenuAnheften          ; an das Window
\M movem.l (sp)+,d5-d6/a2-a6
    rts


Menu_1   dc.b '  Farben',0
         even
FarbText dc.b '   ',0        ; drei Leerzeichen
         end


Listing 1: Sechzehn Farben im Griff
```

```
****************************************************************
*                        AMIGA DOS INFO                        *
*  Programmname          : Listing 2                           *
*  Funktion              : Farbe                               *
*  Sprache               : Assembler                           *
*  Besonderheiten        : Mit A68K erstellt                   *
****************************************************************

FarbRegisterInit ; des neu geoeffneten Screen
*---------------
    movem.l d2-d4/a2,-(sp)  ; benutzte Register
    lea.l   NeueFarben,a2   ; zeigt auf Farbwerttabelle
    move.w  #15,d4          ; 16 Register initialisieren
\farbLoop
    move.l Screen(globals),a0
    lea.l   sc_viewPort(a0),a0 ; Adr Viewport holen
    movem.w (a2)+,d0-d3     ; Register u. RGB in ParaReg
    CallGraph SetRGB4        ; Farbwert setzen
    dbra d4,\farbLoop        ; falls nicht fertig
    movem.l (sp)+,d2-d4/a2
    rts


NeueFarben      dc.w 15,15,15,15 ; Reg. r  g  b  weiss
                dc.w 14,15,00,00 ; d0   d1 d2 d3 rot
                dc.w 13,15,10,00 ; orange
                dc.w 12,15,15,00 ; gelb
                dc.w 11,08,14,00 ; hellgruen
                dc.w 10,02,12,00 ; dunkelgruen
                dc.w 09,00,00,15 ; blau
                dc.w 08,00,06,13 ; dunkelblau
                dc.w 07,09,01,15 ; violett
                dc.w 06,15,01,15 ; dunkelviolett
                dc.w 05,15,10,12 ; rosa
                dc.w 04,13,11,09 ; hellbraun
                dc.w 03,10,08,07 ; dunkelbraun
                dc.w 02,12,12,12 ; hellgrau
                dc.w 01,09,09,09 ; grau
                dc.w 00,00,00,00 ; schwarz


Listing 2: Jetzt kommt eigene Farbe ins Bild
```

```
****************************************************************
*                        AMIGA DOS INFO                        *
*  Programmname          : Listing 3                           *
*  Funktion              : erzeugt Rahmen                      *
*  Sprache               : Assembler                           *
*  Besonderheiten        : Mit A68K erstellt                   *
****************************************************************

eidcmp_Flags = DeltaMove¦MouseMove¦MouseButtons

ZeichnenVorbereiten ; legt IDCMP-Flags auf den Stack
*------------------
    movea.l RastPort(globals),a1 ; Zeichenfarbe setzen
    move.w  FarbNr(globals),d0   ; 1 - 15
    CallGraph SetAPen
```

```
    movea.l RastPort(globals),a1 ; Zeichenmodus best.
    move.b #jam1,d0               ; mit einer Farbe
    CallGraph SetDrMd

    move.l (sp),-(sp)             ; copy Ruecksprungadr.
    move.l wd_IDCMPFlags(our_Win),4(sp) ; merken
    move.l wd_IDCMPFlags(our_Win),d0
    ori.l #idcmp_Flags,d0         ; Flags ergaenzen
    movea.l our_Win,a0
    CallIntui ModifyIDCMP         ; Flags aendern
    rts

ZeichnenAbschliessen ; holt IDCMP-Flags vom Stack
*-------------------
    move.l 4(sp),d0 ; IDCMP zuruecksstellen
    movea.l our_Win,a0
    CallIntui ModifyIDCMP
    move.l (sp)+,a0
    addq.l #4,sp
    jmp    (a0)


Listing 3: Rahmen fuer unsere Zeichenroutinen
```

```
*********************************************************
*                     AMIGA DOS INFO                    *
*  Programmname      : Gummiband                        *
*  Funktion          : zeichnet Linien                  *
*  Sprache           : Assembler                        *
*  Besonderheiten    : Mit A68K erstellt                *
*********************************************************

ZeichneLinie ; zz
*-----------
    bsr ZeichnenVorbereiten
\encore                           ; warten bis 1.But. gedrueckt
    bsr LeftButtonPressed
    tst.b d0
    beq.s \encore

    move.w wd_MouseX(our_Win),d0
    move.w wd_MouseY(our_Win),d1
    movem.w d0-d1,-(sp)           ; Anfangspkt. merken
    movem.w d0-d1,-(sp)           ; vorl.Endpkt. merken
    movea.l RastPort(globals),a1
    CallGraph WritePixel          ; Anfangspunkt setzen

    movea.l RastPort(globals),a1 ; Zeichenmodus best.
    move.b #complement,d0         ; invertieren
    CallGraph SetDrMd
\z
    bsr LeftButtonPressed         ; Abfragen lin. Knopf
    tst.b d0                      ; gedrueckt ?
    bne.s \f                      ; ja, somit fertig

    movem.w 4(sp),d0-d1           ; Startpos holen
    movea.l RastPort(globals),a1
    CallGraph Move                ; Cursor setzen

    movem.w (sp),d0-d1            ;  Endposition holen
    movea.l RastPort(globals),a1
    CallGraph Draw                ; Linie loeschen

    movem.w 4(sp),d0-d1
    movea.l RastPort(globals),a1
    CallGraph Move

    move.w wd_MouseX(our_Win),d0 ; neue Position holen
    move.w wd_MouseY(our_Win),d1
    movem.w d0-d1,(sp)            ; Endpkt. merken

    movea.l RastPort(globals),a1
    CallGraph Draw                ; invert. Linie ziehen
    bra.s \z
\f
    movea.l RastPort(globals),a1 ; Zeichenmodus best.
    move.b #jam1,d0               ; mit einer Farbe
    CallGraph SetDrMd

    movem.w 4(sp),d0-d1   ; Startpos
    movea.l RastPort(globals),a1
    CallGraph Move

    movem.w (sp),d0-d1   ; Startpos
    movea.l RastPort(globals),a1
    CallGraph Draw

    addq.l  #8,sp                 ; Positionen runter
    bsr ZeichnenAbschliessen
    rts

Listing 4: Gummiband und Linie, ein Paint Box Tool
```

```
*********************************************************
*                     AMIGA DOS INFO                    *
*  Programmname      : Listing 5                        *
*  Funktion          : erzeugt Menue                    *
*  Sprache           : Assembler                        *
*  Besonderheiten    : Mit A68K erstellt                *
*********************************************************

ZeichenMenuAnlegen
*-----------------
    bsr WeiteresMenuInit
    beq.s \a
    MenuBenennen Menu_2
    bsr ErstesMenuItemInit
    beq.s \a
    ItemTextEintragen Item20,Cmd20
    bsr WeitererTextInit
    beq.s \a
    bsr WeiteresItemInit
    beq.s \a
    ItemTextEintragen Item21,Cmd21
    bsr WeitererTextInit
    beq.s \a
    bsr WeiteresItemInit
    beq.s \a
    ItemTextEintragen Item22,Cmd22
\a rts


Menu_2  dc.b '  Zeichnen',0
Item20  dc.b ' loeschen',0
Cmd20   =     'n'
Item21  dc.b ' freih.',0
Cmd21   =     'f'
Item22  dc.b ' Linie',0
Cmd22   =     'i'

Listing 5: Menue zur Steuerung des Zeichenmodus
```

```
*********************************************************
*                     AMIGA DOS INFO                    *
*  Programmname      : Listing 6                        *
*  Funktion          : Menue-Auswertung                 *
*  Sprache           : Assembler                        *
*  Besonderheiten    : Mit A68K erstellt                *
*********************************************************

Menu2Auswerten ; Item auf dem Stack
*-------------
    move.w 4(sp),d0 ; ItemNr holen
    bne.s \1        ; 0 nicht, evtl. 1
    bsr LoescheHintergrund
    bra.s \e

\1 cmp.w #1,d0      ; Item 1 ?
    bne.s \2        ; auch nicht
    bsr FreeDraw
    bra.s \e

\2 cmp.w #2,d0      ; eventuell 2 ?
    bne.s \e
    bsr ZeichneLinie

\e move.l (sp)+,a0 ; Ruecksprungadr. holen
    addq.l #2,sp    ; Stack berichtigen
    jmp    (a0)     ; aufruf. Prog. fortsetz.
    rts


\M2
    cmp.w    #2,MenuNr          ; Menu2 angewaehlt ?
    bne.s    \M3
    move.w   ItemNr,-(sp)       ; Parameteruebergabe
    bsr      Menu2Auswerten
    bra      control0           ; Nachricht abholen

Listing 6: Auswertung des "Zeichnen"-Menues
```

```
*********************************************************
*                     AMIGA DOS INFO                    *
*  Programmname      : Listing 7                        *
*  Funktion          : Brushes                          *
*  Sprache           : Assembler                        *
*  Besonderheiten    : Mit A68K erstellt                *
*********************************************************

DrawImage       = -114 ; (rastport,image,leftoffs,rightof
fs)
*                        (a0,a1,d0,d1)
* Image

 struktur
  word   ig_leftEdge
  word   ig_topEdge
  word   ig_width
  word   ig_height
  word   ig_depth
  aptr   ig_imageData
```

```
 byte   ig_planePick
 byte   ig_planeOnOff
 aptr   ig_nextImage
struktLaenge ig_size

###############################################

* Modul ImageStuff

  SECTION ImageStuff,CODE,CHIP

 include "incl/Exec.i"
 include "incl/Types.i"
 include "incl/Intuition.i"

 PUBLIC IntuiBase      ; Importe
 PUBLIC RemKey         ; aus dem Hauptprog.
 PUBLIC Pinsel1
 PUBLIC Pinsel2
 PUBLIC Brush
 PUBLIC kein_Speicher

 PUBLIC PinselInit    ; Export

* Registergebrauch

* global benutzte, d.h. Importe aus dem Hauptprog.

globals    equr a5
our_Return equr d7

MemBlockAnfordern ; Blockgoesse in d0
*-----------------    Speicherliste in RemKey(globals)
   move.l #kein_Speicher,our_Return ; Fehlercode setzen
   move.l RemKey(globals),a0         ; Zugriff Mem Liste
   move.l #Mem_Clear,d1              ; Speichertyp
   CallIntui AllocRemember
   beq.s \M
   move.l a0,RemKey(globals)         ; listenzugriff
\M rts

Pinsel1Init ; Initialisierung der ersten brush
*-----------
   move.l #ig_size,d0
   bsr    MemBlockAnfordern
   beq.s  \e
   move.l d0,Pinsel1(globals) ; Adr Image global merken
   move.l d0,Brush(globals)   ; default Pinsel setzen
   move.l d0,a0
   move.w #1,ig_width(a0)
   move.w #1,ig_height(a0)
   move.w #4,ig_depth(a0)
   lea.l  Pinsel,a1
   move.l a1,ig_imageData(a0)
\e rts

Pinsel2Init ; Initialisierung der ersten brush
*-----------
   move.l #ig_size,d0
   bsr    MemBlockAnfordern
   beq.s  \e
   move.l d0,Pinsel2(globals) ; Adr Image global merken
   move.l d0,a0
   move.w #6,ig_width(a0)
   move.w #4,ig_height(a0)
   move.w #4,ig_depth(a0)
   lea.l  Pinsel,a1
   move.l a1,ig_imageData(a0)
\e rts

PinselInit ; saemtliche Brushes initialisieren;
*----------
   bsr Pinsel1Init
   beq.s \f
   bsr Pinsel2Init
\f rts

   SECTION Images,DATA,CHIP
Pinsel
   dc.w $FFFF,$FFFF,$FFFF,$FFFF

   end

Listing 7: Brushes, bei uns Pinsel, sind Images
```

```
*************************************************************
*                        AMIGA DOS INFO                     *
*  Programmname        : Listing 8                          *
*  Funktion            : Pinsel-Menue                       *
*  Sprache             : Assembler                          *
*  Besonderheiten      : Mit A68K erstellt                  *
*************************************************************

pinselFlags = itemText!highBox!itemEnabled!checkIt!commS
eq

PinselMenuInit ; pp
*-------------
   bsr WeiteresMenuInit
   beq.s \a
   MenuBenennen Menu_3
   bsr ErstesMenuItemInit
   beq.s \a
   move.w #pinselFlags,mi_flags(item) ; Eigenschaft.
   move.l #$FFFE,mi_mutualExclude(item) ; nur ich
   ItemTextEintragen Item30,Cmd30
   bsr WeitererTextInit
   beq.s \a
   bsr WeiteresItemInit
   beq.s \a
   rol.w  mi_mutualExclude+2(item) ; nur ich, kein ander
er
   ItemTextEintragen Item31,Cmd31
   move.l me_firstItem(menu),item ; Adresse erstes SubIt
em
   ori.w #checked,mi_flags(item)  ; Haeckchen setzen
\a rts


Menu_3  dc.b '  Pinsel',0
Item30  dc.b '    fein',0
Cmd30   =        '1'
Item31  dc.b '   mittel',0
Cmd31   =        '2'

Listing 8: Den Pinseln ein eigenes Menue
```

```
*************************************************************
*                        AMIGA DOS INFO                     *
*  Programmname        : Listing 9                          *
*  Funktion            : Pinselauswahl                      *
*  Sprache             : Assembler                          *
*  Besonderheiten      : Mit A68K erstellt                  *
*************************************************************

Menu3Auswerten ; Item auf dem Stack
*-------------
   move.w 4(sp),d0 ; ItemNr holen

   bne.s \1          ; 0 nicht, evtl. 1
   move.l Pinsel1(globals),Brush(globals)
   bra.s \e

\1 cmp.w #1,d0      ; Item 1 ?
   bne.s \e          ; auch nicht
   move.l Pinsel2(globals),Brush(globals)

\e move.l (sp)+,a0 ; Ruecksprungadr. holen
   addq.l #2,sp     ; Stack berichtigen
   jmp    (a0)      ; aufruf. Prog. fortsetz.
   rts

\M3
   cmp.w   #3,MenuNr         ; Menu2 angewaehlt ?
   bne     control0          ; nein, Nachricht abholen
   move.w  ItemNr,-(sp)      ; Parameteruebergabe
   bsr     Menu3Auswerten
   bra     control0          ; Nachricht abholen


Listing 9: Bei zwei Pinseln faellt die Wahl nicht schwer
```

```
*************************************************************
*                        AMIGA DOS INFO                     *
*  Programmname        : Listing 10                         *
*  Funktion            : Zeichenroutine                     *
*  Sprache             : Assembler                          *
*  Besonderheiten      : Mit A68K erstellt                  *
*************************************************************

FreeDraw
*-------
   bsr ZeichnenVorbereiten
\encore                      ; warten bis 1.But. gedrueckt
   bsr LeftButtonPressed
   tst.b d0
   beq.s \encore
\1                           ; Beginn Zeichenschleife
   bsr LeftButtonPressed ; Abfragen lin. But. gedrueckt
   tst.b d0
   bne.s \f                  ; ja, somit fertig

   move.w wd_MouseX(our_Win),d0 ; Mauspos. holen
   move.w wd_MouseY(our_Win),d1
   movea.l RastPort(globals),a0
   move.l Brush(globals),a1
   move.w FarbNr(globals),ig_planePick(a1)
   CallIntui DrawImage
   bra.s \1
\f
   bsr ZeichnenAbschliessen
   rts

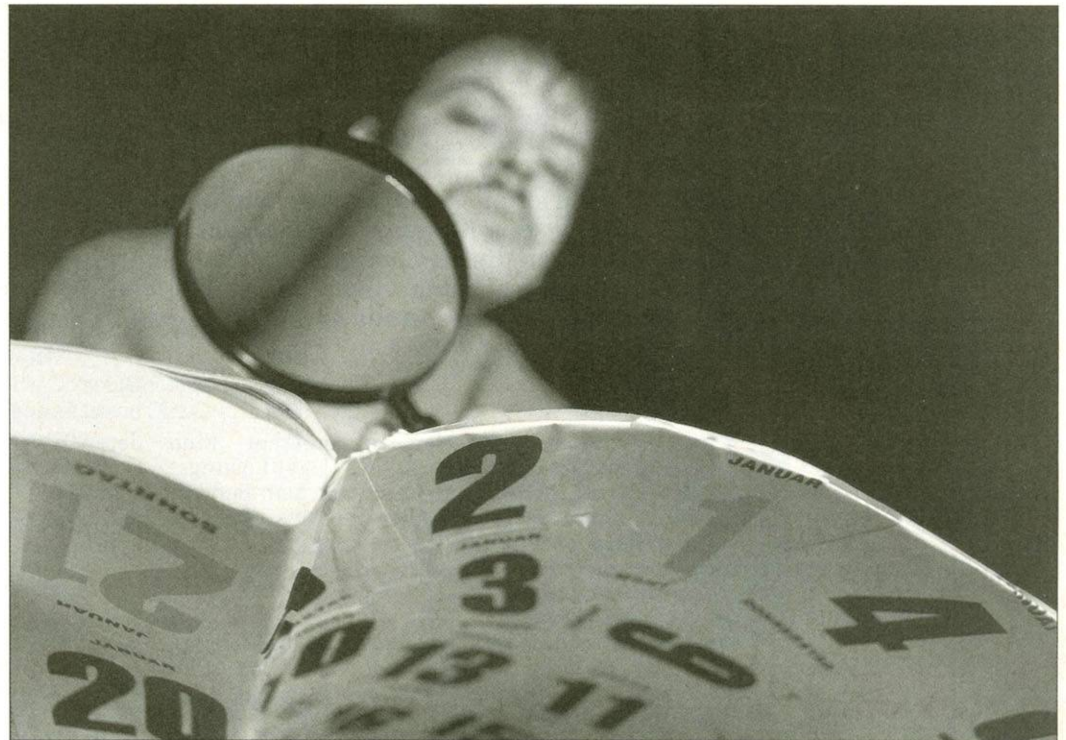
Listing 10: Fuer die Arbeit mit Images eine ueberarbeitete
            Zeichenroutine
```

Bernd Rudolf

Gregor

Berechnung von Wochentagen

Was für ein Wochentag war der 20.5.1962? Halt, bevor Sie nun nach alten Kalendern suchen, sollten Sie sich unser kleines GFA-BASIC-Programm Gregor anschauen, das Ihnen bei der Wochentagsbestimmung zuverlässig weiterhelfen kann.



Die Voraussetzung für die Berechnung von Wochentagen ist, daß das zu berechnende Datum in der Zeitspanne zwischen dem 15.10.1582 (Beginn des Gregorianischen Kalenders) und einem vordefinierten Datum liegt (in unserem Falle ist dies der 01.01.2000, ein Samstag). Diese Definition befindet sich am Anfang des Programms und kann beliebig verändert werden.

Entscheidend für die Berechnung ist die Schaltjahrregelung. Sie besagt, daß alle vier Jahre ein Schalttag eingeschoben wird, ausgenommen den durch 100 teilbaren Jahren, sofern sie nicht durch 400 teilbar sind. Alles klaro? So war zum Beispiel das Jahr 1900 kein Schaltjahr, im Jahr 2000 wird jedoch ebenso wie im Jahr 1996 ein Schalttag eingeschoben.

Doch nun zur Vorgehensweise des Programms: Ausgehend vom Jahr 2000 wird rückwärts Jahr für Jahr der Wochentag des 1. Januars bestimmt, bis das eingegebene Jahr erreicht ist. Es ist unbedingt darauf zu achten, daß bei der Eingabe die volle Jahreszahl verwandt wird (zum Beispiel: 1989)! Ist das gesuchte Jahr gefunden, so wird, wiederum ausgehend vom 1. Januar, der Wochentag des gewünschten Monats bestimmt. Danach muß nur noch der Tag ermittelt werden, und die Arbeit ist getan. Wichtig ist, daß sich ein Schalttag erst nach dem 29.2. eines Jahres bemerkbar macht!

Sicherlich ist es einfacher, bei der Datumsberechnung auf vorgegebene Algorithmen zuzugreifen, jedoch soll ja auch der Lerneffekt nicht zu kurz kommen. Aus diesem Grunde haben wir auch die etwas umständliche Lösung des Problems in Angriff genommen. Wer ein wenig Spaß an Gregor gefunden hat, kann sich ja einmal daransetzen und das Programm ein wenig modifizieren.

Wie immer gilt natürlich auch bei diesem Listing: Zeilennummern nicht mit eintippen, die dienen nämlich nur der Übersicht.

(br)

Listings

```
REM ****************************************
REM *                 Gregor               *
REM *  (c) 1989 by B. Rudolf & AMIGA DOS   *
REM ****************************************
tag=1
monat=1
jahr=2000
wtag=6
REPEAT
  INPUT "Tag   : ",etag
  INPUT "Monat : ",emonat
  INPUT "Jahr  : ",ejahr
UNTIL etag<32 AND emonat<13 AND ejahr>1581
AND ejahr<2000
FOR j=jahr-1 DOWNTO ejahr
  IF (j/400=INT(j/400)) OR (j/4=INT(j/4)) THEN
    wtag=wtag-2
    schalt=1
  ELSE
    wtag=wtag-1
    schalt=0
  ENDIF
  IF j/100=INT(j/100) THEN
    wtag=wtag+1
    schalt=0
  ENDIF
  IF wtag<1 THEN
    wtag=wtag+7
  ENDIF
NEXT j
```

Listing Gregor

```
RESTORE monat
t=0
FOR m=1 TO emonat-1
  READ a
  t=t+a
NEXT m
t=t+etag-1
IF schalt=1 AND emonat>2 THEN
  t=t+1
ENDIF
wtag=wtag+(t MOD 7)
IF wtag>7 THEN
  wtag=wtag-7
ENDIF
IF schalt=0 AND emonat=2 AND etag>28 THEN
  PRINT "Den ";etag;".";emonat;".";ejahr;
  PRINT " gibt es nicht!"
  GOTO ende
ENDIF
RESTORE wochentage
FOR i=1 TO wtag
  READ wtag$
NEXT i
PRINT "Der ";etag;".";emonat;".";ejahr;
PRINT " ist ein ";wtag$
ende:
END
w$=INPUT$(1)
monat:
DATA 31,28,31,30,31,30,31,31,30,31,30,31
wochentage:
DATA Montag,Dienstag,Mittwoch,Donnerstag,Freitag
DATA Samstag,Sonntag
```

Listing Gregor

Die Tüftelecke

"Was brauch' ich denn einen CD, hab' doch einen Amiga", hört man in letzter Zeit immer häufiger aus den Kehlen der Amiganer, was die eindrucksvollen musikalischen Qualitäten dieses Computers nur unterstreicht. Wir wollen diesem Umstand natürlich Rechnung tragen und stellen den Wettbewerb dieses Monats unter das Motto "Wenn Amiga singt, donn legst di nieder..."

Den diesmonatigen anspruchsvollen Preisen wollen wir eine nicht ganz so einfache Aufgabe vorausstellen. Wir haben sieben Fragen vorbereitet, die jeweils drei Antwortmöglichkeiten besitzen. Jede Antwort ist mit drei Buchstaben versehen, von denen nur einer für das letztlich zu ermittelnde Lösungswort zutreffend ist. Welcher von den dreien das ist, nun, das sollen Sie selber herausfinden. Puzzlen ist angesagt. Eine kleine Hilfe: Das Lösungswort stammt aus dem Bereich 'Computer und Musik'

Der Gewinn

Zu verlosen gibt es diesmal fünf Sidmon-Soundeditoren und spannende Spiele der Firma Turtle Byte. Teilnahmeberechtigt sind alle, sofern sie nicht Mitarbeiter des DMV-Verlages sind oder sich deren Angehörige nennen. Der Rechtsweg ist ausgeschlossen.

Schicken Sie Ihre Lösung an den:

DMV-Verlag
Redaktion AMIGA DOS
Stichwort "Musik"
Postfach 250
3440 Eschwege

Einsendeschluß ist der

15. März 1990. (jf)

Frage 1: Was ist ein Intervall?
- ein musikalisches Zwischenspiel — s-t-u
- der Abstand zwischen zwei Tönen — a-b-c
- Sturz in frischen Strassenbelag — f-g-h

Frage 2: Was versteht man unter dem Begriff Riff?
- Rock im fier/fiertel-Takt — w-x-y
- wiederholter Melodieverlauf — a-b-c
- Karibische Reggaevariation — m-n-o

Frage 3: Wann wurde das Saxophon erfunden?
- im 19. Jahrhundert — k-l-m
- im 20. Jahrhundert — t-u-v
- das gab es immer schon — c-d-e

Frage 4: Ist Fuzz
- ein in die Jahre gekommener Fuzzi? — i-j-k
- ein Bassbooster? — f-g-h
- ein Verzerrer? — b-c-d

Frage 5: Kraut Rock nennt man
- deutsche Rockmusik der 70er Jahre — k-l-m
- ein besonders unästhetisches Kleidungsstück — x-y-z
- diffuse Dreiklangstrukturen — t-u-v

Frage 6: Terz...
- ...ist ein jazzig-dissonantes Trio — q-r-s
- ...gibt es, wenn man zu laut Musik hört — e-f-g
- ...ist die dritte Stufe einer diatonischen Tonfolge — c-d-e

Frage 7: Der Musikprozessor des Amigas heisst
-- Paula — r-s-t
-- Markus — b-c-d
-- Nena — h-i-j

Spiele für Amiga

SPEICHERERWEITERUNG AMIGA 500

auf 1 MB, inklusive Uhr und DUNGEON MASTER (engl.)
nur 279,00DM

Wir bekommen täglich Neuheiten für den Amiga! Fordern Sie doch einfach mal die neueste Preisliste an, Sie werden überrascht sein! Hier ein paar Beispiele:

TITEL	PREIS	TITEL	PREIS
Axel's Magic Hammer	59,90	Great Courts (Tennis)	74,80
Battle Squadron	74,80	Fighter Bomber	89,80
Beverly Hills Cop	74,80	It came from the Desert	84,80
Chase HQ	74,80	Kaiser	109,00
Clown-o-Mania	54,80	Little Computer People	34,80
Day of the Viper	74,80	Treasure Island Dizzy	19,80
Dogs of War	59,90	Twinworld	74,80
Drakkhen	89,00	Weird Dreams	79,80

X-Copy II + Cyclone + Hardware nur 69,00DM

Versandkosten:

5,00 DM bei Warenwert unter 100,00 DM
Warenwert über 100,00 DM Versandkosten frei!

COMPY SHOP

Gneisenaustr. 29
4330 Mülheim Ruhr
0208-497169 + 496178

JEDES TEIL WOANDERS EINKAUFEN ?
BEI SCHAEFER BRAUCHST DU NUR EINMAL LAUFEN !

AMIGA-FLACHBETT-SCANNER — DM 948,00
Kann als BILDERFASSUNGSGERÄT/Kopierer und Thermodrucker eingesetzt werden. Scanndichte 200 Punkte/Zoll, Scannzeit 10 Sekunden. Ablage des ganzen Bildes im IFF. Auflösungen 320x200 - 640x512 werden unterstützt. Ausschnittvergrößerungen möglich. Binär u. 16 Grau-Darstellungen.

VIDEOTEXT-DECODER — DM 298,00
Man kann das VIDEOSIGNAL eines Recorders, Scart-TV oder Tuners dazu verwenden, den freien Service des Teletextes im IFF- oder ASCII-Format abzulegen. Super-Grafik-Darstellung.

ZWEITLAUFWERK "Senator 3,5"	DM 259,00
MIDI-MASTER/Midi-Intervace (Amiga-Typ angeben)	DM 99,00
MIDI-MUSIC-MANAGER (siehe Eurosystems)	DM 49,00
MIDI-Master und MUSIK-MANAGER, zusammen	DM 120,00
PRO-SAMPLER und DATEL-JAMMER	DM 169,00
GENIUS-MAUS - voll Amiga-kompatibel	DM 79,50
BURST-NIBBLER inkl. neueste Hardware	DM 149,00
SYNCRO.EXPRESS inkl. neueste Hardware - Weiterentwicklung des Burst-Nibbler bei Benutzung externer Laufwerke -	DM 149,00
GFA-BASIC 3.0	DM 198,00
DEVPACK-ASSEMBLER (M+T-Verlag)	DM 149,00
AMIGA-DEUTSCH - Deutsche DOS-Befehle einlesen	DM 29,95
SPRACHENKIT - Vokabel-Trainer in Deutsch, Latein, Engl., Französ., Span., Italien., Japanisch !	DM 49,95

AMIGA-Ersatzteilpreise wegen starker Schwankungen derzeit nur auf Anfrage.

Alle Bücher von Markt und Technik - umfangfreiche Software, Hardware und Ersatzteile auf Anfrage. - Versand nur gegen Vorkasse + DM 5,00 oder Nachnahme + DM 8,00.

(CLS)-COMPUTERLADEN SCHAEFER
Klingelholl 111, 5600 Wuppertal-2, Tel.: 0202/508121
Geschäftszeiten: Mo.+Di., Do.+Fr. 14-18.30 Uhr, Samstags 10-13 Uhr

---

Camp-Gewinnspiel 1990


Jörg Finger

Computer-World verlost 3 Wochen Computerferien

Die Preisfrage:
We heißt der Skateboard-Champion und Vizeweltmeister AM 1989 im Freestyle, der auf unseren Computer- und Sportcamps 1990 als Trainer dabei ist?

So könnt Ihr gewinnen:
Schreibt einfach den gesuchten Namen auf eine Postkarte und sendet diese bis 31. Mai 1990 an Computer World.

Teilnahmebedingungen:
Teilnehmen kann jeder, außer den Mitarbeitern der Fa. Computer World EDV-Ausbildung & Computerferien GmbH und deren Angehörige. Einsendeschluß ist der 31. 5. 1990. Die Gewinner werden benachrichtigt. Der Rechtsweg ist ausgeschlossen.

Computercamps 1990:

Freiburg ◇ Feldberg ◇ Utzenfeld

Computing:
BASIC ◇ GFA-BASIC ◇ Turbo-Pascal
Maschinensprache ◇ 'C'-Programmierung
PC + MS-DOS ◇ PC-Anwendung ◇
Desktop-Video

Computersysteme:
AMIGA ◇ ATARI ST ◇ C 64 ◇ C 128
IBM-kompatible PC's ◇ CPC 6128/464

American-Sports:
skateboard ◇ bmx ◇ football ◇ frisbee
baseball ◇ mountain bike ◇ bumerang

Reiten:
Computerkurs direkt auf dem Reiterhof

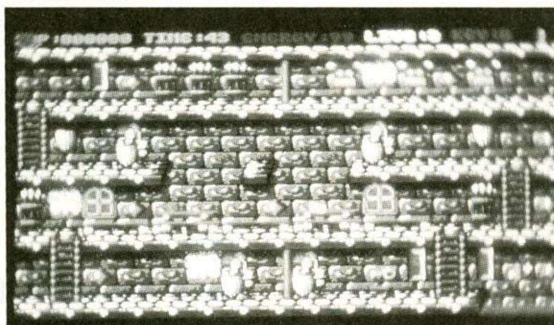
Freizeit:
Panorama-Rundflug ◇ Sommerrodelbahn
Europa-Park ◇ Tennis ◇ Volleyball
Kino ◇ Schwimmen ◇ Billard ◇ Disco

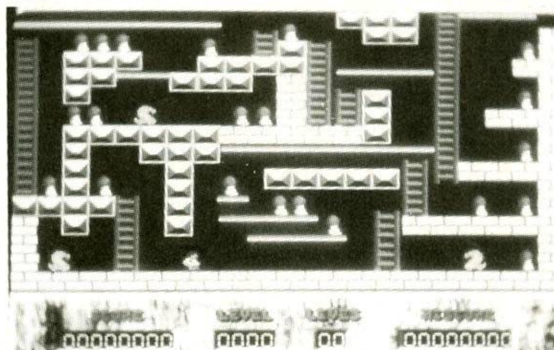
Sofort kostenlosen Katalog anfordern!

Hurstweg 62 B · 7800 Freiburg i. Br.
Telefon 0761/447 75

Computer World

Kunert-Soft, eine Abteilung der Firma Computer-Express in Essen, hat sich ein Ziel gesteckt, daß dem Amigabesitzer zugute kommen soll.
Angefangen mit dem Amigaprogramm "Money-Player Deluxe", einem Geldspielgerät, der einem echten Automaten in nichts nachsteht. Sie können sogar Ihren Gewinn auf Diskette oder Festplatte speichern. Walzen starten, Walzen stoppen und Gewinne riskieren ist natürlich im Programm auch enthalten.
Unser zweites Programm, mit dem Namen "Speedrunner" ist ein Geschicklichkeitsspiel, mit über 240 Levels und einem komfortablen Leveleditor. Ein Spiel, daß Sie wahrscheinlich für Wochen oder Monate an Ihren Amiga fesseln wird, mit sehr guter Grafik und Stereosound.
Bei unserm dritten Programm, auch ein Geschicklichkeitsspiel, mit dem Namen "Danger-Castle" (gefährliches Schloß) ist uns ein Game gelungen, daß fast keine Wünsche mehr offen läßt. Sie müßen viele Räume durchsuchen und die Diamanten einsammeln, aber Vorsicht, überall lauern verschiedene Energiefresser auf Sie oder der Boden unter Ihnen verschwindet. Wenn Sie glauben, Sie können die Levels bewältigen ohne genau nachzudenken, wohin Sie laufen, werden Sie nicht weit kommen. Auch wenn Sie einmal verloren haben, haben Sie dreimal die Möglichkeit, mit dem letzten erreichten Level neu anzufangen. Auch bei diesem Spiel wird Ihr Highscore mit abgespeichert.
Das waren drei Beispiele aus unserer Softwarepalette, die immer größer wird. Wir verfügen über (eigene) Programmierer für neue Programme von Supergames bis hin zu komfortablen Anwenderprogrammen, kaufen jedoch auch gute Software für den Amiga gern an. Natürlich zu fairen Preisen oder Konditionen. Ein Amigaprogramm muß nicht 139,- DM kosten. Wir wollen alle Orginal-Software für maximal 39,- DM an den Amigabesitzer bringen. Daher verzichten wir auf teure Anleitungen bzw. Verpackungen. Eine einfache Anleitung bringts auch. Immerhin kommt es auf das Programm an, da wird natürlich nicht gespart.
Kunert-Soft können Sie bei Computer-Express, Gladbecker Str. 6 in 4300 Essen 1 direkt oder bei gut sortierten Softwareanbietern bestellen. Jedes Kunert-Soft-Orginal kostet nur 39,- DM





COMPUTER-EXPRESS

Inh. D. Gnoth — ***Gladbeckerstraße 6***
4300 Essen 1 — ***Tel.: 0201 - 31 24 59***



Kontakte zu schaffen zwischen der Redaktion und den Lesern, das ist das Ziel der Leserbriefseiten. Nachdem wir in der letzten Ausgabe noch keine Briefe veröffentlichen konnten (zu einer Zeitschrift, die es noch nicht gibt, kann man sich nicht äußern), liegen uns nun die ersten kritischen Leserstimmen und Anregungen vor, die wir natürlich zur Diskussion stellen wollen. Doch wir erwarten nicht nur Ihre kritischen Zuschriften, auch mit Fragen, Problemen, Antworten oder Tips & Tricks sollten Sie nicht hinter dem Berg halten.

SIE FRAGEN, WIR ANTWORTEN!

Keine Farbe im PC-Teil

Sehr geehrte Damen und Herren, ich besitze einen Amiga 2000 und habe diesen kompatibel gemacht (PC Board und VGA, XT 8088 und 8-Bit-VGA-Karte).
Mein Problem ist nun, wenn ich ein Spiel auf dem 'PC' laden will, erscheint dieses nur in Schwarzweiß.
Was könnte ich bei der Hardware-Installation verkehrt gemacht haben? Oder, viel wichtiger, welche Icons und Befehle muß ich in der 'Workbench' eingeben, um auf dem PC Farbe zu bekommen? Und gibt es eine Möglichkeit auf meinem Amiga PC-Sound zu bekommen?

R. Wettermann
Wilburgstetten

Nun ja, der Amiga 2000 ist auch MS-DOS-tauglich. Die Betonung liegt dabei aber auf 'auch'. Der Unterschied liegt ganz einfach darin, daß viele Spiele des MS-DOS-Systems auf die Grafikkarten und die Soundausgabe des PCs zugeschnitten sind. Die Grafik- und Soundausgabe bei PCs ist jedoch derart unterschiedlich zum Amiga, daß es gerade bei professioneller Software zu Problemen kommen kann, in Ihrem Fall fehlen Farbe und Sound. Der Amiga emuliert an sich nur die CGA-Grafikkarte. Um auf der PC-Karte farbige Grafiken darstellen zu können, benötigen Sie das Amigafile PCCOLOR der Workbench bzw. der PC-Bridgeboard-Diskette. Wenn Sie die MS-DOS-Seite damit starten, können Sie die farbige Seite der PC-Software kennenlernen. Sobald der DOS-Prompt erscheint, müssen Sie nur noch MODE CO80 eintippen.
Wenn die PC-Seite Sounds unterstützen soll, so müßten Sie eine kleine Hardwarebastelei vornehmen, da die Lautsprecher nur von der Amigaseite unterstützt werden. Wollen Sie bessere Ergebnisse erzielen, werden Sie um einige Neuanschaffungen wie externe EGA-Grafikkarte oder die AdLib- Soundkarte (beide für die PC-Steckplätze) nicht herumkommen.

(Red.)

Mehr Speicher für den A500

Hallo AMIGA-DOS-Team! Endlich eine Zeitschrift, die auch Neueinsteigern eine echte Chance gibt (wohl weil Ihr neu seid, was?). Macht weiter so, denn ich habe euch schon weiterempfohlen. Deswegen blamiert mich nicht! Doch nun zu meinem Anliegen: Ich habe seit kurzem einen Amiga 500 und möchte diesen unter Umgehung der 'teuren' Speichererweiterungen, mit Einbeziehung der 512-kByte-Erweiterung, auf 2,5 MByte aufrüsten. Habt Ihr einen Tip oder eine Bauanleitung auf Lager? Basteln könnte ich selbst.

Alfred Brühl
Siegenburg-Kaldauren

Erst mal ein Dankeschön für das Lob (auch Redakteure leben nicht nur von Luft), und das mit dem Blamieren, das wollen wir uns ja selbst nicht antun.
Eine Bauanleitung für eine 2-MByte-Erweiterung ist keine schlechte Idee, nur haben wir leider im Augenblick keine vorrätig. Den einzigen Tip, den ich Ihnen geben kann, ist der: Einige Hardwareanbieter haben eine interne Speichererweiterung bis 2 MByte im Angebot. Diese wird in den Memoryschacht gesteckt (wie die 512-kByte-Erweiterung). Die Kosten betragen zirka 700 DM, sie können durch die Tagespreise der RAM-Bausteine variieren. In der Redaktion ist ein 500er mit einer internen, abschaltbaren 2-MByte-Erweiterung bestückt, wir möchten sie gar nicht mehr missen.
Einige Firmen bieten die Platinen zu den RAM-Cards auch unbestückt mit Bauplan an, die Platinen kosten je nach Bauart 60,- bis 200,- DM (bis 8 MByte bestückbar), hier sollten Anzeigen aufmerksam gelesen werden.
Die Idee mit der 'RAM-Karte im Eigenbau' bleibt auf jeden Fall schon mal im Gedächtnis.

(Red.)

3,5 Zoll unter MS-DOS

Vor wenigen Tagen habe ich einen Amiga 2000 und eine PC-Karte mit einem 5 1/4-Zoll-Laufwerk gekauft. Von der PC-Seite her möchte ich nun gerne auch auf Programme auf 3 1/2-Zoll-Disketten zurückgreifen. Daher folgende Frage: Gibt es eine Möglichkeit, auf das zweite, interne 3 1/2-Zoll-Laufwerk wahlweise von der Amiga-Seite oder von der PC-Seite zuzugreifen?

Lothar Selenz
Köln

Sie können die Diskettenlaufwerke der Amiga-Seite nur für den Austausch von ASCII-Dateien (Texte) benutzen. Dafür stehen die PC-Files AREAD und AWRITE zur Verfügung. Kopieren Sie sich diese beiden Programme auf eine MS-DOS-Diskette. Nun starten Sie die PC-Seite und aktivieren das Amiga-PC-Icon.
Wollen Sie beispielsweise eine Amiga-Textdatei Text1 von Laufwerk df1: auf die PC-Seite kopieren, so gehen Sie wie folgt vor:
AREAD df1:Text1 a:text1
Andererseits können Sie auch ASCII-Files von der PC-Seite auf den Amiga konvertieren. Dies sähe in unserem Beispiel so aus:
AWRITE a:Text2 df1:Text2

(Red.)

Druckerprobleme und Lesefehler

Ich habe seit zirka vier Monaten einen Amiga 500. Zwischenzeitlich habe ich auch einige Zusatzgeräte erworben. Seit einiger Zeit habe ich jedoch Probleme, die ich alleine nicht lösen kann.

Das erste Problem bezieht sich auf den Drucker. Es ist ein Star LC-10. Als Druckertreiber benutze ich den Standard-Epson-Treiber auf der Workbench 1.3. Der Textausdruck funktioniert, wie Sie sehen, einwandfrei. Zu schaffen macht mir lediglich der Grafikausdruck. Bis jetzt ist es mir noch nicht gelungen, Grafiken in NLQ-Qualität auszudrucken. Ist es überhaupt möglich, den Star LC-10 als NLQ-Grafikdrucker zu nutzen und falls ja, was mache ich falsch?

Nun zu meinem zweiten Problem. Seit einiger Zeit habe ich auf verschiedenen Disketten sehr häufig Read/Write-Errors. Betreffende Disketten habe ich auch schon mit 'Virus X 3.2' kontrolliert. Es wurde allerdings kein Virus entdeckt. Kann es möglich sein, daß es einen neuen Virus gibt, der 'Read/Write-Errors' erzeugt, aber von 'Virus X 3.2' nicht erkannt wird, oder liegt hier ein anderer Fehler vor? Das Problem haben allerdings auch einige Bekannte von mir auf ihren Amigas.

Thomas Emmerich
Gelsenkirchen

Zu Problem 1: NLQ bezieht sich auf den Textausdruck bei 9-Nadel-Druckern (Near Letter Quality = Fast Schreibmaschinenschrift). Trotzdem müßte Ihr Drucker ohne weiteres in der Lage sein, Grafiken auszudrucken. Zuvor sollten Sie in Preferences einen der externen Druckertreiber (auf der EXTRAS-Diskette zu finden) eintragen, hier kann zum Beispiel EPSON LX... der Druckertreiber sein. Außerdem müssen Sie im Druckerauswahl-Window den Druck auf 'GREY SCALE' setzen. 'COLOR' ist nur für Farbdrucker (wie den LC-10 Colour) nützlich.

Der Grafikausdruck läßt sich am besten mit einem Grafikprogramm testen. Hier können Sie nämlich die Werte für die beste Farbumsetzung in Graustufen herausfinden.

Vielleicht auch ein Tip zu entsprechender Software: In dieser Ausgabe stellen wir das Druckprogramm TurboPrint von Irsee-Soft vor, das erstens über eine Menge Druckertreiber verfügt und zweitens dem Anwender die Möglichkeit für viele Einstellungen vorgibt.

Zu Problem 2: Sie haben sich höchstwahrscheinlich den 'Lamer Exterminator II'-Virus eingefangen. Dieser schreibt den Sektor einer Diskette mit der ASCII- Zeichenfolge 'LAMER!' voll, worauf diese beim Lesen einen READ-/WRITE-ERROR abgibt. Auf der Time__5-PD-Diskette ist der VIRUSX 4.0 zu finden, der dem Lamer-Exterminator den Garaus macht.

Bild verfärbt

Seit etwa vier Monaten tritt an meinem Amiga 500 (Kickstart Version 33.180 / WB. 1.2) folgendes auf: Nach dem Einschalten erscheint das typische Bild (Hand mit Diskette), bei mir einmal schwarz verfärbt oder mal blau, dann wieder mit Strichen verschiedener Größe und Farbe.

Manchmal hilft der Griff zur Paniktaste (Ctrl + A + A), manchmal helfen auch zehn Resets hintereinander nicht weiter.

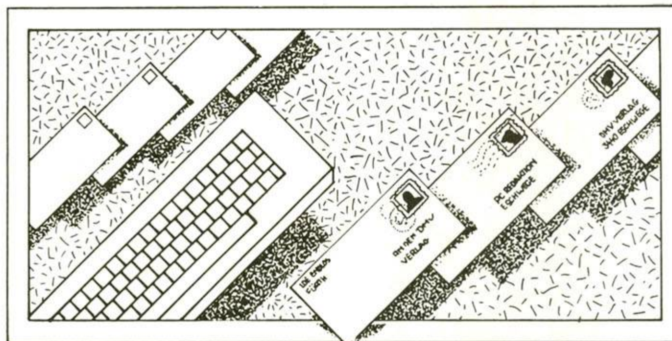
Schon zweimal hatte ich, als er wieder streikte, den Deckel abgeschraubt und alles so weit abgebaut, daß ich an den ROM-Baustein heran konnte. Den hebelte ich vorsichtig heraus und steckte ihn dann wieder zurück.

Nach dem Zusammenbau (Garantie schon abgelaufen), schaltete ich den Computer wieder ein und, oh Wunder, es war wieder alles normal. Nun kann ich aber nicht jedesmal den Rechner auseinanderpflücken, denn das bekäme ihm wahrscheinlich nicht so gut. Ich hatte schon den Verdacht, daß Temperaturschwankungen (Dauerlüften) dieses hervorrufen könnten, aber auch dieser Verdacht entkräftete sich. Wenn Sie eine Erklärung für das Phänomen wissen, lassen Sie es mich wissen.

Martin Wellmann
Wuppertal

So, wie es aussieht, haben Sie Probleme mit dem Grafik-Chip, hier liegt höchstwahrscheinlich der Defekt vor. Integrierte Schaltungen haben die unangenehme Eigenschaft, Defekte erst nach bestimmten Voraussetzungen zu zeigen, entweder durch Erhitzung oder Fehlschaltungen. In Ihrem Fall ist es besser, den Amiga einmal zur Überprüfung in eine Service-Werkstatt zu geben. Auf keinen Fall sollten Sie ohne Kenntnisse irgendwelche Bauteile aus den Sockeln nehmen, hierbei kann es sogar noch zu größeren Defekten kommen.

(Red.)



DoubleStrike und der Fehlerteufel

Ich habe gestern Ihre erste Ausgabe gekauft (AMIGA DOS 1/90) und bin begeistert. Jetzt meine Frage: Ich habe gleich die Abtipphilfe Hextype eingegeben. Nach dem Start wird nach dem Namen des 'Hexdump-Files' gefragt, dann nach den Zeilen und der Länge. Sie schreiben, 'diese Information erfahren Sie in den ersten drei Zeilen des Listings'. Den Namen habe ich gefunden, aber die Zeilen und die Länge? Was muß ich eingeben? Zeilen – anhand der Zeilennummer der letzten Listingzahl? Die Länge – in Bytes oder was? Bitte machen Sie mich doch etwas schlauer.

Klaus Jürgen Hampel
Sibbesse

Wie schön, daß wir alle nicht perfekt sind, auch die Redaktion muß ab und zu Asche über ihr Haupt streuen. Bei den Hexdump-Listings sind aus Versehen die falschen Listingköpfe abgedruckt worden. Richtigerweise muß da stehen:

```
Name : DoubleStrike
Zeilen : 540
Länge : 8660
Name : DSGraph
Zeilen : 350
Länge : 6706
```

Das Ganze läßt sich aber auch sichtbar machen anhand der Zeilennummern vor den Zeilen. Wenn man die mit der Anzahl der Bytes pro Zeile multipliziert, bekommt man die Länge.

Übrigens hat sich beim Drucken des Listings in der Druckerei ein Wert umgebildet, der bei uns im Listing noch richtig war:

In Zeile 315 des ersten Listings muß das elfte (11.) Byte nicht 8C heißen, sondern 80.

In einigen Heften war aus der 0 eine Ecke herausgefallen, so daß diese nicht mehr als solche zu erkennen war. Wir bitten, die Fehler zu entschuldigen.

'Gefrorene' Bilder

Seit einem halben Jahr bin ich Besitzer eines Amiga 500, zweier Laufwerke, einer Speichererweiterung und seit 14 Tagen eines Druckers.

Ich habe das Malprogramm 'The Graphic Studio' und kann damit selbstgemalte und Bilder von den Demodisketten ausdrucken. Ich habe nun eine PD-Diskette mit sehr schönen Bildern, die hintereinander erscheinen. Wie kann man die Bilder einfrieren und ausdrucken?

Heinz Peter Pesch
Mönchengladbach

Was Sie brauchen, ist ein sogenannter Software-Freezer, der Bilder auf Diskette abspeichert. Da gibt es nur ein Problem: Die meisten uns bekannten Programme aus Public-Domain oder Shareware machen ihren Job nur ungenügend, die hinterher ausgedruckten Bilder sind meistens kaum als solche zu erkennen, vor allem, wenn es um Hi-Res-Bilder geht. Als Tip kann man hier wieder das TurboPrint-Programm nennen, was aber als kommerzielles Programm etwas teurer wird. Zweitens könnte das Modul 'Pro-Access' helfen, welches in der Lage ist, Bilder aus Programmen auf Diskette abzuspeichern, es ist allerdings ebenfalls keine preiswerte Lösung.

Hier müssen wir ein bißchen an die Leser appellieren: Mel-

Eine Bitte an unsere Abonnenten

Vermerken Sie bei Schriftverkehr und Zahlungen neben der vollständigen Anschrift stets Ihre Abo-Nummer.

Sie vermeiden damit unnötige Verzögerungen bei der Bearbeitung Ihres Abonnements.

Vielen Dank

Ihre DMV-Versandabteilung

den Sie sich, falls Ihnen ein Programm aus der PD bekannt sein sollte, das Bilder im Rechner einfriert und diese auch noch in guter Qualität an den Drucker schickt, Herr Pesch und die Redaktion werden es Ihnen danken.

(Red.)

Lob und Tadel

Zunächst einmal möchte ich Euch zu Eurem Magazin beglückwünschen.
Ich wünsche mir die AMIGA DOS als anspruchsvolle Zeitschrift, die gewisse Schwerpunkte setzt, und nicht als Zeitschrift, die versucht, jedem gerecht zu werden.
Computerbastelei – Sicherlich gibt es genug Elektronikzeitschriften. Aber warum soll man sich seine Artikel für den Amiga aus X Zeitschriften heraussuchen? Darum ein dickes JA für Basteleien.
Programmierung/Listing – Ja und Nein. Programmierkurse und Projekte (z.B. Context Amiga) und in diesem Zusammenhang anschauliche und notwendige Listings JA. Aber bitte keine ellenlangen Spielelistings.
Kurse – Ja!!! Noch und nöcher. Assembler, Grafik, Sound, Hardware, DFÜ. Aber bitte keine BASIC-Kurse mehr.
Software- und Hardwaretests – Tests müssen sein (man muß sich schließlich irgendwie orientieren), aber bitte kurz und prägnant.
Tips&Tricks – ...sind immer gut, auch und besonders für Einsteiger.
Workshops – Halte ich für uninteressant. Ein Workshop kann meist nicht allgemein genug sein, da ist ein gutes Handbuch schon besser.
Unterhaltung – Spiele sind eine feine Sache, aber inzwischen gibt es genug Zeitschriften mit riesigem Spieleteil. Ich finde, diese Seiten sollten in der AMIGA DOS besser genutzt werden.
Public Domain – Hat meiner Meinung nach große Beachtung verdient.
DFÜ, BTX – Diese Bereiche halte ich für zukunftsträchtig, sie sollten mehr beachtet werden.
DATABOX – Ehrlich gesagt halte ich den Preis von 24,- DM für unverschämt, auch wenn es zwei Disketten sind. Wie wäre es, wenn Ihr eine Mailbox einrichten oder BTX-Anbieter würdet und so die Programme für einen geringen Unkostenbeitrag anbieten würdet?
Sonstiges – Gut gemachte Grundlagenartikel, z.B. Compilerbau oder was Euch sonst alles einfällt.

Thomas Helbing
Erkelenz

Den Geschmack aller zu treffen, da haben sich schon Meisterköche vergeblich dran versucht. Aber uns freut, daß Ihnen das Heft gefallen hat. Zu den einzelnen Punkten jetzt Stellung zu nehmen, hieße, den Rahmen des Heftes zu sprengen. Nur soviel:
Daß AmigaBASIC ein ungeliebtes Kind ist (teils zu Recht, teils zu Unrecht), darüber läßt sich nicht streiten, jedoch haben die Amigabesitzer diese Sprache als erstes auf ihrem Computer vorgefunden. Und daß man in AmigaBASIC auch schöne Sachen programmieren kann, wenn man ein paar Regeln beachtet, steht außer Frage. Und deshalb wollen wir den Amiga-Einsteigern, die nicht sofort mit Assembler, Modula oder C beginnen, ein wenig mit dem BASIC-Kurs unter die Arme greifen.
Der Unterhaltungsteil ist ein wichtiges Gebiet auf dem Amiga. Gerade durch seine Sound- und Grafikfähigkeiten ist er eine gute Anlaufstelle für Spieler. Deshalb werden wir auch in Zukunft die Spiele als festen Bestandteil der AMIGA DOS betrachten.
Über die DATABOX direkt ist wohl nur zu sagen, daß AMIGA-DOS-Leser hier die Gelegenheit bekommen, die veröffentlichten Programme lauffähig zu beziehen. Eine Verbreitung über DFÜ oder Btx ist im Augenblick noch nicht im Gespräch.

(Red.)

Hextype und DoubleStrike

Sehr geehrte Damen und Herren, kurz vor Weihnachten stöberte ich in unserem Computerladen herum. Da viel mein Augenmerk auf das Zeitschriftenregal, und siehe da, ein neues Gesicht. Nachdem ich diesem Neuling einen Teil meiner Zeit gewidmet hatte, entschloß ich mich, die 'Studien' zu Hause fortzusetzen. Schon nach dem ersten Durchblättern hatte ich einen guten Eindruck. Als ich aber das ganze Heft durchgeblättert hatte, war ich wie vor den Kopf geschlagen: Überzeugende Gliederung, alle wichtigen Gebiete wie Grafik, Sound und Spiele behandelt, Workshops, Listings – und alles für 6,50 DM. Das war spitze.
Um einen besseren Eindruck zu bekommen, setzte ich mich hin und begann Double Strike mit dem HexType einzugeben. Leider lief das Ganze noch nicht (Unable to load Double Strike: Error Code 121). Trotzdem habe ich mich von der Arbeit des HexType überzeugen können. Alles in allem ein Bombenheft. Hier nun noch ein paar Verbesserungsvorschläge:
– Kennzeichnen Sie im Inhalt die Programme, die auf der DATABOX enthalten sind. Das ist besser als die Doppelseite im Heft, die leicht als Werbung überblättert wird.
– Die Vorzüge des Hex-Dumps liegen auf der Hand: Kürzere Eingabezeit, weniger Platzbedarf im Heft. Es ist jedoch nicht von der Hand zu weisen, daß diese Art Listings schädlich und ermüdend für die Augen ist. Sourcecodes wären lehrreicher.
– Zu HexType an sich: Sie schreiben, die geforderten Daten zum Eingeben des Listings befinden sich in den ersten drei Zeilen des HexDumps. Das kann ich nicht bestätigen, zumal mir die Funktion der Variablen 'DLEN' erst klar wurde, als es bereits zu spät war. Hier bietet es sich doch an, diese Informationen in dem sowieso schon vorhandenen Info-Kästchen unterzubringen.

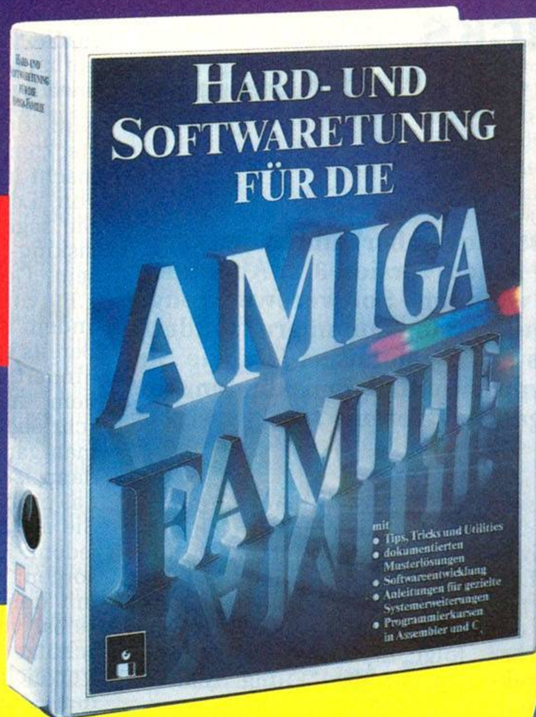
Christian Möller
Stade

Zu DoubleStrike und seinen fehlenden Informationen: Siehe oben!
Die Verbesserungsvorschläge werden wir überdenken, zum DATABOX-Info sind wir schon fast überredet, das Ganze zieht sich allerdings noch etwas hin. (Wenn Sie dieses Heft in Händen halten, ist der Februar angebrochen, dann sind wir von der Redaktion schon bei Heft 5.)
Hexdumps werden wir nur dann abdrucken, wenn es unumgänglich ist, also der Source-Code Kataloggröße erreicht. Ansonsten gilt: Wer lernen will, soll Quelltext abtippen!

(Red.)

START...

... in neue Amiga-Dimensionen





... in neue 68000er-Dimensionen

Amiga-Tuning durch Hardware-erweiterungen, Systemprogrammierung, Tips und Tricks

Dieses neue Nachschlagewerk erläutert Ihnen übersichtlich und detailliert, wie Sie durch gezieltes Tuning die Fähigkeiten Ihres Amiga erweitern und optimal ausschöpfen.

Aus dem Inhalt:

- **Lernen Sie die Bausteine Ihres Amiga genau kennen:** In einem eigenen Hardwarekapitel erschließen Sie sich den Aufbau Ihres Amiga. Der umfassenden Beschreibung der Motorola-CPU und der Customchips Agnus, Paula, Denise und Gary entnehmen Sie Architektur, Register und Befehlssatz sowie die Abläufe bei Adressierung und Befehlsausführung.
- **Amiga-Tuning durch Hardwareerweiterungen:** Für alle, die gern mit dem Lötkolben hantieren, hält das Werk präzise Bauanleitungen mit Platinenlayout, Schaltbild und Bestückungsplan für preiswerte Selbstbau-Erweiterungen bereit, z.B. Sound-Digitizer, EPROM-Brenner.
- **Sie haben die Wahl: Basic, C, Assembler** ... In Kursen eignen Sie sich eine ausgefeilte Programmiertechnik an. Mit den Übungsbeispielen verfügen Sie gleichzeitig über Module für eigene Programme.
- **Praxiserprobte Musterlösungen zum Sofort-einsatz,** u.a. zu Desktop-Publishing und Genlock-Anwendungen, Dateiverwaltung, Meßwerterfassung und -verarbeitung.
- **3D-Grafik – Animation – Video: lassen Sie Ihrer Kreativität freien Lauf!** Erschließen Sie sich das breite Spektrum der Amiga-Grafik: von einfachen geometrischen Körpern über 3D bis hin zur Animation.
- **Software-Engineering** ist gefragt: Hinweise zum Einbinden von Intuition-Funktionen für Menüs, Alerts od. Systemrequester ermöglichen Ihnen, den Anwenderkomfort Ihrer Programme zu erhöhen.

Fordern Sie noch heute an:

Hard- und Softwaretuning für die Amiga-Familie

praktischer Ringbuchordner, DIN A4, Grundwerk ca. 480 Seiten, inkl. Diskette, Bestell-Nr.: 4800/74800, Preis: DM 98,–

Alle 2 – 3 Monate erhalten Sie Ergänzungsausgaben zum Grundwerk mit je ca. 130 Seiten, Seitenpreis 38 Pfennig (Abbestellung jederzeit möglich).

Perfektes Anwender-, System- und Bausteinprogrammieren in Assembler und C

Was bisher nur Minis und Mainframes vorbehalten war, schaffen jetzt auch Mikrocomputer mit 680xx CPUs. Nutzen Sie diese Vorzüge! Das neue 68000er-Handbuch vermittelt Ihnen hierzu detailliertes Hardwarewissen, fundierte Betriebssystemkenntnisse und ausgefeiltes Programmier-Know-how in Assembler und C.

Die wichtigsten Themen auf einen Blick:

- **Detaillierte Hardwarebeschreibungen** der 68000er-Prozessorfamilie: Die Prozessoren 68000, 68008, 68020 und 68030 sowie die 8-, 16-, 32-Bit-Peripheriebausteine werden ausführlich beschrieben.
- **Vertieftes Betriebssystem-Know-how:** Sie lernen die Strukturen der 68000er-Betriebssysteme (wie GEM-TOS, OS-9, Unix und Amiga-DOS) kennen.
- **C-Programmierung:** In C erstellen Sie schnelle und kompakte Programme und bedienen sich zugleich einer Programmiersprache, die sich bestens zur Systemprogrammierung eignet.
- **Programmierkurs Assembler:** Durch maßgeschneiderte Befehlsfolgen, die direkt die Hardware und das Betriebssystem ansprechen, wird optimaler Code generiert.
- **Tools und Utilities** inkl. einem universellen symbolischen Disassembler zur Bearbeitung fremder Programme, die nicht im Quellcode vorliegen.
- **Bewährte Musterlösungen** wie zum Erstellen von 3D-Grafiken, Auswertung statistischer Daten.
- **Systemtuning durch Hardwareerweiterungen:** Ausführliche Anleitungen zum sicheren Nachbau z.B. von Ein-/Ausgangsinterfaces für Atari ST oder DFÜ-Modem für Amiga.

Fordern Sie noch heute an:

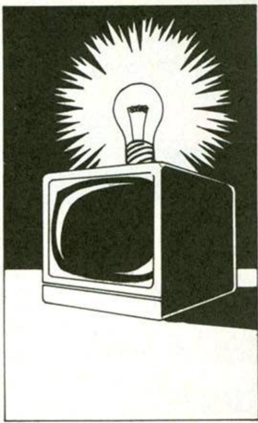
Erfolgreiches Programmieren von 68000er Systemen in Assembler und C

praktischer Ringbuchordner, DIN A4, Grundwerk ca. 550 Seiten, Bestell-Nr.: 3400, Preis: DM 92,–

Alle 2 – 3 Monate erhalten Sie Ergänzungsausgaben zum Grundwerk mit je ca. 130 Seiten, Seitenpreis 38 Pfennig (Abbestellung jederzeit möglich).



INTEREST-VERLAG
Fachverlag für
Special Interest Publikationen
und Anwendersoftware
Industriestraße 21 · D-8901 Kissing
Telefon 0 82 33/21 01-0



Zum Editor (m2emacs): schnellere Arbeitsweise, erheblich verbessertes Scrollen von Texten, eingebauter File Requester, Compilieren aus dem Editor heraus möglich, umschaltbar auf Interlace.

Edgar Meyzis

Tips & Tricks, nicht nur für M2Amiga

(Teil 3)

Seit der Amiga-Expo 89 (November) gibt es den Compiler M2Amiga in der neuen Version 3.3. Aber auch die dazugehörigen, weiteren Softwarewerkzeuge tragen nicht nur höhere Versionsnummern, sondern haben sich gleichfalls stark gemausert [1], wie Sie den folgenden Stichworten entnehmen können, die wesentliche Neuerungen skizzieren.

M2 Amiga hebt ab

Zum Compiler (m2c): residentfähig (damit entfallen Ladezeiten), Schnittstellenmodule (zum Betriebssystem und zu Libraries) sehr gründlich überarbeitet und auf den Stand von Kickstart 1.3 bzw. Workbench 1.3 gebracht, mehrere zusätzliche Compileroptionen (z.B. Ausrichten globaler Variablen auf Langwortadressen, Laden von Code ins "Chip Ram"), wesentlich effizientere "Stack-Überprüfungen", Quelltext zu sämtlichen Modulen einschließlich des Laufzeitsystems erhältlich, zusätzliche Codeoptimierungen, verbessertes Laufzeitsystem und leichtere Einbindung von Maschinencode.
Zum Linker (m2l): residentfähig, erfragt auf Wunsch für jedes Modul, welcher Bibliothek es entnommen werden soll.
Zum Make (m2make): Ausstattung mit einem Modullader, der den Programmstart ohne zu linken ermöglicht, Optionen für die Arbeitsweise des Compilers spezifizierbar.
Zum Debugger (m2r): umschaltbar auf Interlace, Größe und Lage der fünf Ausgabefenster abspeicherbar und als Knüller: Datentypen sind redefinierbar, um z.B. "generische" Typen (wie ADDRESS) als Zeiger zu interpretieren und ein "Durchhangeln" durch bzw. Auswerten von Strukturen zu ermöglichen.
Zur Programmierumgebung (m2apse): lädt Compiler und Linker resident, unterstützt sämtliche Werkzeuge des M2Amiga-Paketes in vorbildlicher Weise.
Greifen wir nun aus dem DOS des Amiga die "Lock-Prozeduren" heraus und streifen dabei "Open" und "Close". Wir wollen Ihnen weder einreden, daß die "Lock-Prozeduren" etwas mit dem Liebesleben von Walrossen zu tun haben, noch wollen wir die Prozeduren im einzelnen beschreiben oder gar die Struktur von "Locks" darstellen. Dafür ist schon genug Druckerschwärze aufgewandt worden (z.B. in [2] und [3]). Statt dessen wollen wir ein wenig über die Auswirkungen von "Locks" berichten sowie ein Modul vorstellen, das das "Locking" sichtbar macht.

Von offenen und versteckten Locks

Was sind eigentlich "Locks"? Sprachlich gesehen sind "Locks" Verbindungen, Kopplungen bzw. Verriegelungen. Auf den Amiga bezogen bieten "Locks" eine besondere Form des Zugriffes auf Dateiverzeichnisse bzw. auf Dateien über die jeweilige Blocknummer (Laufwerke sind blockweise organisiert). Da die Blöcke fortlaufend numeriert sind, ergeben sich eindeutige, absolute Zugriffe. Verzeichnisse und Dateien verweisen mit Blocknummern aufeinander. Daraus folgt: Ein "Lock" ist als Einstieg in das "Filing System" geeignet.

"Locks" sind in zwei unterschiedlichen Ausprägungen, "sharedLock" und "exclusiveLock" einsetzbar. Die erste Ausprägung ist die allgemeine, die wenig Restriktionen für andere "Zugreifer" auf das betroffene Objekt schafft. Das "exclusiveLock" hingegen blockiert die davon betroffene Datei (bzw. das Verzeichnis) vor Zugriffen, z.B. durch ein Programm. "sharedLocks" können beliebig häufig für ein und dasselbe Objekt des "Filing System" hergestellt werden. "exclusiveLocks" hingegen dulden keine Nebenbuhler. Sie tolerieren es auch nicht, wenn ein Objekt bereits mit einem "sharedLock" belegt sind, und verweigern ihre Aufgabe. Und wie merkt sich DOS die erhobenen "Locks"? Sie ahnen es schon. Natürlich wird von einer einfach verketteten Liste Gebrauch gemacht. DOS entnimmt ihr die vorhandenen "Locks" in der jeweiligen Ausprägung.

Nun kommen wir zu den versteckten "Locks". Das DOS macht intern regen Gebrauch von "Locks". Na klar, werden Sie sagen. Bei einem Multitasking-System muß das so sein. Geöffnete Dateien sind z.B. vor dem Zugriff durch andere Prozesse zu schützen. Bereits ein "sharedLock" verhindert, daß eine Datei von einem anderen Prozeß überschrieben werden kann. Will man gar das Lesen verbieten, dann ist ein "exclusiveLock" das probate Mittel. Aber nicht Sie als Anwender müssen in jedem Fall für die "Locks" sorgen. Das Amiga-DOS setzt mit den Prozeduren "Open" selbst verschiedene "Locks", schützt somit Dateien bzw. Verzeichnisse vor unpassenden Zugriffen. Damit sind hervorragende Voraussetzungen für eine "Multi-User-Fähigkeit" gegeben, die bei der Vernetzung von Amigas zu "Cluster" bald gefragt sein könnte.

Bei der Anwendung von "Open" spielen die qualifizierenden Argumente "newFile", "readWrite" und "oldFile" eine entscheidende Rolle. Die Argumente bestimmen indirekt, mit welcher Ausprägung von "Locks" DOS intern zu arbeiten hat.

Mit der Tabelle wird der Versuch unternommen, die Verriegelungsmechanismen zu veranschaulichen. Die linke Spalte stellt die jeweils erste, erfolgreich verlaufene Aktion ("Lock" oder "Open") mit den unterschiedlichen, qualifizierenden Argumenten dar.

Step 1	Step 2				
	File Modes			Lock Types	
erfolgreich	newFile	oldFile	readWrite	shared	exclusive
newFile	NIL	NIL	NIL	NIL	NIL
oldFile	NIL	ok	NIL	ok	NIL
readWrite	NIL	NIL	NIL	NIL	NIL
shared	NIL	ok	NIL	ok	NIL
exclusive	NIL	NIL	NIL	NIL	NIL

Tabelle: Offene und versteckte Locks (Step 1) lassen nur begrenzte weitere Aktionen (Step 2) zu

In der Kopfzeile sind mögliche Folgeaktionen auf Objekte des "Filing System" aufgeführt, deren erfolgreicher Verlauf ein "ok" in der Matrix bezeichnet. Aus der ersten Zeile geht hervor, daß ein Öffnen einer Datei mit dem Argument "newFile" intern ein "exclusiveLock" setzt, somit keine weiteren Zugriffe auf die Datei zuläßt. Zeile zwei bietet die Information, daß Dateien beliebig häufig zum Lesen (oldFile) geöffnet werden dürfen. Schwache "sharedLocks" sind gleichfalls zugelassen. Eine mit "readWrite" geöffnete Datei gebärdet sich genauso intolerant wie eine mit "newFile" angelegte. Ein "sharedLock" als erste Aktion läßt nur das Lesen der betroffenen Datei sowie weitere "schwache Locks" zu.
Damit dürfte die Verriegelungstaktik des DOS klar sein, und man könnte das Heft spätestens jetzt gelangweilt aus der Hand legen, wenn "Locks" vom Laufzeitsystem bei Programmbeendigung genauso entgegenkommend behandelt werden würden wie ungeschlossene Dateien. Leider bleiben "Locks" in der erwähnten, systeminternen Liste erhalten, sofern sie nicht mit "UnLock" ausgetragen werden.
Aber wie fehlerhafte Anwendungen aufspüren? Listing 1 und Listing 2 enthalten das Werkzeug dazu. Es werden rein äußerlich die üblichen "Lock-Prozeduren" zur Verfügung gestellt. Listing 2 jedoch offenbart, daß zusätzlicher Code erzeugt wird, der Ausgaben auf dem Bildschirm produziert, um die "Lock-Vorgänge" zu beschreiben.
So sind z.B. für die Variable "puffer" keine besonderen Vorkehrungen zu treffen, um sie an einer Langwortadresse zu plazieren. Das Modul "Dos" wurde insgesamt importiert, um Namenskonflikte mit den angebotenen "Lock-Prozeduren" auszuschließen. Die Ausgaben des Moduls erfolgen in unterschiedlicher, farblicher Darstellung. Die Prozedur "UnLock" testet zunächst auf "NIL", um einem (Ab)sturz Ihrer(s) Geliebten vorzubeugen.

"Look" auf PRT: ?!

In der Prozedur "Lock" wird der Systemaufruf "Examine" verwendet, der als Parameter ein "Lock" benötigt, um Zugriff auf "File Header" zu erhalten, die über die Struktur "FileInfoBlock" (gleichfalls automatisch auf Langwortadresse ausgerichtet) zugänglich sind.
Mit einem "sharedLock" können wir leicht feststellen, ob ein Objekt des "Filing System" existiert. DOS bietet uns jedoch kein Mittel zu erfassen, ob ein Drucker ansprechbar ist. Listing 4 bietet Ihnen einen Weg, sich vom Einschaltzustand Ihres Druckers zu überzeugen. Dazu müssen wir "PRT:" im Gegensatz zur vorherigen Betrachtungsweise als Device auffassen. Wir schlagen Ihnen vor, daß Sie den angebotenen Code Ihren Bedürfnissen anpassen und in ein handliches externes Modul verpacken.

(mm)

Literaturhinweise:

[1] Gute Nachrichten Nr.3, Berichte, Anwendungen und Neuigkeiten von der Programmiersprache Modula-2

[2] Amiga Intern, Band 2

[3] Amiga-Dos-Handbuch 1.3

Listings

```
*************************************************************
*                     AMIGA DOS INFO                        *
*  Programmname       : Listing 1                           *
*  Funktion           : Modulschnittstelle                  *
*  Sprache            : Modula2                             *
*  Besonderheiten     : keine                               *
*************************************************************

DEFINITION MODULE LockMon;

FROM Dos IMPORT
        FileLockPtr;

TYPE STRING = ARRAY[0..47] OF CHAR;
     STRPtr = POINTER TO STRING;

PROCEDURE CreateDir(name : STRPtr) : FileLockPtr;

PROCEDURE CurrentDir(lock : FileLockPtr) : FileLockPtr;

PROCEDURE DupLock(lock : FileLockPtr) : FileLockPtr;

PROCEDURE Lock(name : STRPtr; accessMode : LONGINT)
                                        : FileLockPtr;

PROCEDURE ParentDir(lock : FileLockPtr) : FileLockPtr;

PROCEDURE UnLock(lock : FileLockPtr);

END LockMon.

Listing 1: Schnittstelle zum Modul, das Aufrufe
           von "Lock-Prozeduren" sichtbar macht
```

```
*************************************************************
*                     AMIGA DOS INFO                        *
*  Programmname       : Listing 2                           *
*  Funktion           : Modul-Implementation                *
*  Sprache            : Modula2                             *
*  Besonderheiten     : keine                               *
*************************************************************

IMPLEMENTATION MODULE LockMon;

IMPORT Dos;
FROM Terminal IMPORT
        Write, WriteLn, WriteString;
FROM SYSTEM IMPORT
        ADDRESS, ADR, INLINE;

TYPE LockProc = (cur, dup, par);

VAR  StrPtr  : STRPtr;
     zlock   : Dos.FileLockPtr;
     info    : Dos.FileInfoBlockPtr;
     done    : BOOLEAN;
     invers,
     normal,
     orange  : ARRAY[0..9] OF CHAR;

     puffer  : Dos.FileInfoBlock;


PROCEDURE WriteInvers(txt : ARRAY OF CHAR);
BEGIN
   WriteString(invers);
   WriteString(txt);
   WriteString(normal);
END WriteInvers;


PROCEDURE WriteDir(txt : ARRAY OF CHAR);
BEGIN
   WriteString(orange);
   WriteString(txt);
   WriteString(normal);
END WriteDir;


PROCEDURE MultiLock(prozedur : LockProc;
                    lock : Dos.FileLockPtr)
                    : Dos.FileLockPtr;
VAR txt : ARRAY[0..17] OF CHAR;
BEGIN
   IF prozedur = cur THEN
      zlock := Dos.CurrentDir(lock);
      txt   := " := CurrentDir(";
   ELSIF prozedur = dup THEN
      zlock := Dos.DupLock(lock);
      txt   := " := DupLock(";
   ELSE (* par *)
      zlock := Dos.ParentDir(lock);
      txt   := " := ParentDir(";
   END;
   done := Dos.Examine(zlock, info);
   WriteDir(info^.fileName);
   WriteString(txt);
   done := Dos.Examine(lock, info);
   WriteString(info^.fileName);  WriteString(");");
   WriteLn;
   RETURN zlock;
END MultiLock;


PROCEDURE CreateDir(name : STRPtr) : Dos.FileLockPtr;
BEGIN
   zlock := Dos.CreateDir(name);
   IF zlock = NIL THEN
      WriteInvers("NIL");
   ELSE
      done := Dos.Examine(zlock, info);
      WriteDir(info^.fileName);
   END;
   WriteString(" := CreateDir(");
   WriteString(name^);
   WriteString(");"); WriteLn;
   RETURN zlock;
END CreateDir;


PROCEDURE CurrentDir(lock : Dos.FileLockPtr)
                                  : Dos.FileLockPtr;
BEGIN
   RETURN MultiLock(cur, lock);
END CurrentDir;
```

```
PROCEDURE DupLock(lock : Dos.FileLockPtr)
                                    : Dos.FileLockPtr;
BEGIN
    RETURN MultiLock(dup, lock);
END DupLock;


PROCEDURE Lock(name : STRPtr; accessMode : LONGINT)
                                    : Dos.FileLockPtr;
BEGIN
    zlock := Dos.Lock(name, accessMode);
    IF zlock = NIL THEN
        WriteInvers("NIL");
    ELSE
        done :=  Dos.Examine(zlock, info);
        IF info^.dirEntryType > 0 THEN (* Directory *)
            WriteDir(info^.fileName)
        ELSE
            WriteString(info^.fileName);
        END;
    END;
    WriteString(" := Lock(");
    WriteString(name^);
    Write(",");
    IF accessMode = Dos.sharedLock THEN
        WriteString("sharedLock");
    ELSIF accessMode = Dos.exclusiveLock THEN
        WriteString("exclusiveLock")
    ELSE
        WriteInvers("ungueltig")
    END;
    WriteString(");"); WriteLn;
    RETURN zlock;
END Lock;


PROCEDURE ParentDir(lock : Dos.FileLockPtr)
                                    : Dos.FileLockPtr;
BEGIN
    RETURN MultiLock(par, lock);
END ParentDir;


PROCEDURE UnLock(lock : Dos.FileLockPtr);
BEGIN
    WriteString("UnLock(");
    IF lock # NIL THEN
        done := Dos.Examine(lock, info);
        WriteString(info^.fileName);
        Dos.UnLock(lock);
    ELSE
        WriteInvers("NIL");
    END;
    WriteString(");"); WriteLn;
END UnLock;


BEGIN
    info := ADR (puffer);
    invers := "_7m"; invers[0] := CHR(9BH);
    normal := "_0m"; normal[0] := CHR(9BH);
    orange := "_0;33;40m"; orange[0] := CHR(9BH);
END LockMon.

Listing 2: Modul zum Aufruf von "Lock-Prozeduren" aus
           Dos, das diese durch eigenen Code umhuellt,
           um ablaufende VorgΣnge darzustellen
```

```
****************************************************************
*                        AMIGA DOS INFO                        *
*  Programmname         : Listing 3                            *
*  Funktion             : Testmodul                            *
*  Sprache              : Modula2                              *
*  Besonderheiten       : keine                                *
****************************************************************

MODULE LockMonTest;

FROM Dos IMPORT
        FileLockPtr, sharedLock;
FROM LockMon IMPORT
        CreateDir, CurrentDir, DupLock, Lock,
        ParentDir, UnLock;
FROM SYSTEM IMPORT
        ADR;


VAR ourLock,
    cd        : FileLockPtr;


BEGIN
    ourLock := Lock(ADR("df1:Trashcan"), sharedLock);
    cd := Lock(ADR("df1:Trashcan"), sharedLock);
    UnLock(cd);
    UnLock(ourLock);
END LockMonTest.

Listing 3: Einfaches Testmodul fuer "LockMon"
```

```
****************************************************************
*                        AMIGA DOS INFO                        *
*  Programmname         : Listing 4                            *
*  Funktion             : Druckertest                          *
*  Sprache              : Modula2                              *
*  Besonderheiten       : keine                                *
****************************************************************

FROM Exec IMPORT
        IOStdReq, IOStdReqPtr, MsgPortPtr, OpenDevice,
        CloseDevice, write, DoIO;
FROM ExecSupport IMPORT
        CreatePort, DeletePort, CreateExtIO, DeleteExtIO
;
FROM Parallel IMPORT
        Status, StatusSet;
FROM Printer IMPORT
        IODRPReq, IOPrtCmdReq, prtCommand, ris, Special,
        SpecialSet, printerName, rawWrite;

VAR prtPort     : MsgPortPtr;
    prtPtr      : IOStdReqPtr;


PROCEDURE ClosePrinter;
BEGIN
    CloseDevice (prtPtr);
    DeleteExtIO (prtPtr);
    DeletePort (prtPort);
END ClosePrinter;


PROCEDURE PrinterOpen() : BOOLEAN;
VAR printerOpen : BOOLEAN;
BEGIN
    prtPort := CreatePort (NIL, 0);
    IF prtPort # NIL THEN
        prtPtr  := CreateExtIO (prtPort, SIZE(IOStdReq));
        IF prtPtr = NIL THEN
            DeletePort (prtPort);
            printerOpen := FALSE;
        ELSE
            OpenDevice (ADR (printerName), 0,
                                    prtPtr, LONGSET{0});
            printerOpen := prtPtr^.error = 0;
            IF ~ printerOpen THEN
                DeleteExtIO (prtPtr);
                DeletePort (prtPort);
            END;
        END;
    END;
    RETURN printerOpen;
END PrinterOpen;


PROCEDURE PrintRaw (text : STRING);
VAR pos : INTEGER;
BEGIN
    pos := Length (text);
    IF pos > 0 THEN
        INC (pos);
        text[pos] := 15C; text[pos+1] := 0C;
        WITH prtPtr^ DO
            command := rawWrite;
            data    := ADR(text);
            length  := pos + 1;
        END;
        DoIO (prtPtr)
    END;
END PrintRaw;

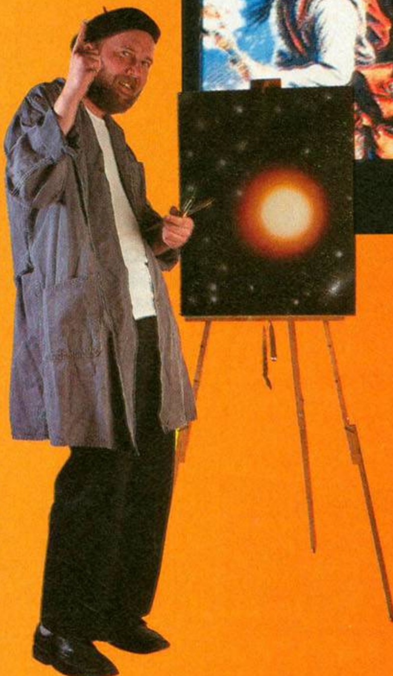

PROCEDURE PrinterReady(text : STRING) : BOOLEAN;
CONST Query = 12;
VAR prtStatus : StatusSet;
    done      : BOOLEAN;
BEGIN
    IF PrinterOpen() THEN
        WITH prtPtr^ DO
            command := Query;
            data    := ADR (prtStatus);
        END;
        DoIO (prtPtr);
        done := NOT (pSel IN prtStatus) AND
                    (pBusy IN prtStatus);
        IF done THEN
            PrintRaw(text)
        END;
        ClosePrinter;
    ELSE
        done := FALSE
    END;
    RETURN done;
END PrinterReady;


Listing 4: "PrinterReady" ermittelt, ob ein Drucker
           ansprechbar ist. Mit "PrintRaw" (optional)
           kann der Drucker z.B. initialisiert werden.
```

AMIGA-DOS-Pixel-Panorama



Legend nennt sich dieses Pixel-Prachtstück, das auf einer PD-Diskette zu finden ist



Grafik und Farbenpower sollten nicht immer nur aus der Konserve kommen, wenn man einen derartigen Grafikgiganten wie den Amiga zu Hause stehen hat. Ob stilvolles Landschaftsbild, fotorealistischer Akt, comicartige Actionszenen oder aufwendige Raytracing-Bilder, die Bilderwelt des Amiga ist schier grenzenlos.

Sicherlich sind schon viele unter Ihnen, liebe Leser, einmal auf die Idee gekommen, ein eigenes Bild mit dem Computer zu erstellen, oder Sie haben sogar schon eigene Werke fertiggestellt. Wie auch immer, wir und alle anderen Leser der AMIGA DOS interessieren sich für Ihre Mal- und Zeichenkünste.

Das Pixel-Panorama möchte Ihre Amiga-Galerie werden. Hier werden wir Ihnen in Zukunft sehenswerte Amiga-Bilder aller Schattierung präsentieren. Dabei werden wir natürlich nicht nur Bilder aus kommerziellen Produkten vorstellen, sondern auch ganz besonders die Bilder unserer Leser, die sich hoffentlich rege an dieser Aktion beteiligen. Als Dankeschön winkt für jede Veröffentlichung ein Softwarepaket im Wert von zirka 100 DM. Alle, die sich beteiligen wollen, sollten uns ihre Meisterwerke, auf einer Diskette abgespeichert, zuschicken. Auf dem Etikett der Diskette sollte neben Ihrer kompletten Anschrift auch noch das Grafikformat bzw. das Grafikprogramm vermerkt sein, mit dem das Bild erstellt wurde. Das Ganze stecken Sie dann in einen Umschlag und schicken es an den

DMV-Verlag
Red. AMIGA DOS
Kennwort Pixel
Postfach 250
3440 Eschwege





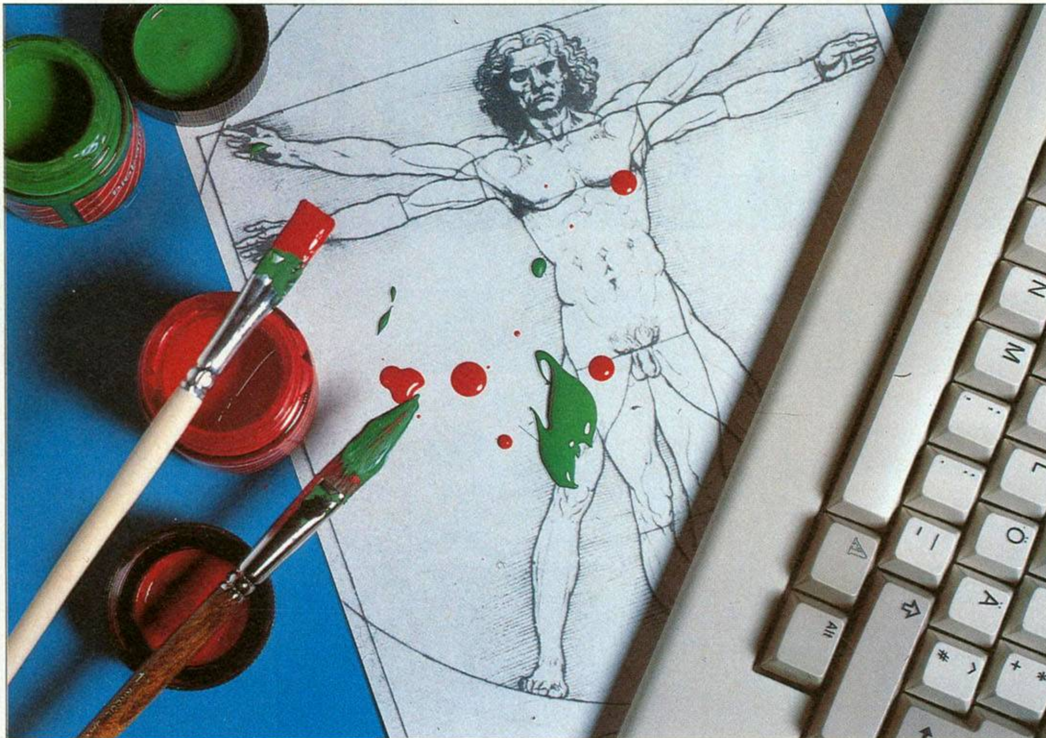
Vampire, Stiere und High-Tech-Vögel. Impressionen aus bekannten Spielen wie Baal und Chrono Quest und einer PD-Grafiksammlung



Wir hoffen auf eine rege Beteiligung aller Leser. Diejenigen unter Ihnen, die ganz und gar nicht malen können, sollten uns ihre Meinung zu dieser Aktion schreiben.

Einstweilen jedoch wünschen wir Ihnen viel Spaß beim Betrachten der Bilder, die wir für diese Ausgabe ausgewählt haben.

(hs)



Mit Deluxe Paint wie ein Profi arbeiten

Farbe und Form – Komponenten, die für den Künstler, der eine Leinwand benutzt, genauso von Relevanz sind wie für den Computerkünstler. Deluxe Paint wartet mit einer Vielzahl von Funktionen und Möglichkeiten auf, die den Anwender durchaus verwirren können.

Nachdem der letzte Teil dieses Kurses sich der nackten Erklärung der Menüs widmete, die der Anwender in der Kopfzeile von Deluxe Paint vorfindet, wollen wir diesmal ein wenig tiefer schürfen und auf die Funktionen näher eingehen, die einer genaueren Erklärung bedürfen. Dabei werden wir ein wenig anders vorgehen als bisher. In diesem Kursteil werden wir darauf verzichten, jeden Menüpunkt durchzukauen. Wir werden nur an jenen Punkten länger verweilen, die einer Erläuterung bedürfen. Punkte, die keine genauere Erklärung verlangen oder die bereits in einem früheren Teil eingehend erklärt wurden, bleiben unberücksichtigt und werden einfach ausgelassen. Um trotzdem eine Orientierung zu gewährleisten, werden Deluxe-Paint-spezifische Bezeichnungen in spitzen Klammern erscheinen, und Menü- bzw. Windowbezeichnungen werden in der Schreibweise übernommen, wie sie in Deluxe Paint gebräuchlich sind.
Wir wählen diese Regelung, da wir in diesem Teil des Kurses viele Menüpunkte einfach überspringen werden. So gibt es z.B. im <Picture>-Menü keinen Punkt, auf den näher eingegangen wird. Die ersten Optionen, die einer Erklärung bedürfen, finden sich im <Brush>-Menü.

Kursfahrplan

Teil 1: Das Basismenü und seine Funktionen

Teil 2: Die Kopfleiste und die dort enthaltenen Menüs

Teil 3: Die Anwendung der in der Kopfleiste enthaltenen Funktionen

Teil 4: Animation mit dPaint III

Teil 5: Arbeitstips vom Profi

Spezialitäten im <Brush>-Menü

Einer der Punkte, die Deluxe Paint derartig leistungsfähig machen, ist die Option, einen eigenen Pinsel zu definieren. Aber es ist mehr möglich, als einen Teil aus einem Bild zu 'schneiden' und diesen dann als Zeichenstift zu benutzen. Diese Bildausschnitte, die im Deluxe-Paint-Jargon als 'Brushes' bezeichnet werden, können vergrößert, verkleinert, gedreht, gebogen oder gekippt werden. Alle Optionen, die sich auf die Größe eines Brushes (wir werden von hier an diesen kurzen Namen für Bildausschnitte verwenden) oder mit der Manipulation einer solchen beziehen, sind in dem Untermenü <Size> zusammengefaßt. Die hier enthaltenen Punkte lassen sich problemlos anwenden. Allerdings gibt es zwei Besonderheiten, die der Anwender wissen sollte. Diese Eigenarten lassen sich am besten an dem Beispiel eines kleinen Kreises erklären. Setzen wir voraus, die Linie, die den Kreis beschreibt, ist an allen Stellen exakt ein Pixel stark. Benutzen Sie nun die Option <Double>, verdoppelt Deluxe Paint Ihren Brush konsequent. Im Klartext, die horizontale und die vertikale Ausdehnung des Brush hat sich verdoppelt, und die Linie, die den Kreis beschreibt, ist nun zwei Pixel stark.
Mittels der Option <Stretch> können Sie einen Brush dehnen, als wäre er ein Bild auf einer flexiblen Gummihaut. Aber denken Sie immer daran, wenn Sie einen Brush vergrößern, wird seine eckige Pixelstruktur mitvergrößert. Dadurch wirken so vergrößerte Brushes immer sehr klobig.
Wird ein Brush hingegen verkleinert, auf welche Art auch immer, verliert er dabei im Detail. Das kann mitunter so weit gehen, daß vom ursprünglichen Motiv nicht mehr viel erkennbar ist.
Eine andere Art, einen Brush zu verkleineren, ist <Trimm>, eine Option aus dem <Edge>-Untermenü. Mit dieser Funktion kann rund um einen Brush eine Reihe Pixel weggenommen werden. Dieser Vorgang kann beliebig oft wiederholt werden, sogar bis von dem Brush nichts mehr übrig ist.
Vergrößert bzw. ergänzt wird ein Brush durch <Outline>. Rund um den Brush herum erscheint eine Reihe Pixel in der Farbe, die in der Palette gerade aktiviert ist.
Im Zusammenspiel von <Trimm> und <Outline> ist es also möglich, einen Brush durch Randlinien in beliebiger Anzahl und Farbe zu vergrößern bzw. durch

Ausblenden je einer Randlinie zu verkleinern. Das <Rotate>-Untermenü und seine Funktionen dienen dem Drehen eines Brushes. Um hier möglichst gute Ergebnisse zu erzielen, sollte bereits beim Definieren des Brushes eine Regel beachtet werden: Schneiden Sie außer dem von Ihnen tatsächlich gewünschten Bildausschnitt nach Möglichkeit so wenig wie möglich 'Bildumfeld' mit aus. Beim Drehen eines Brushes wird nämlich immer die linke, untere Ecke des Brushes als Angelpunkt für die Drehung bestimmt. Insbesondere bei der Benutzung der Rotationsfunktion <Any Angle> können dadurch unliebsame Effekte erzielt werden.

Die Arbeit im <Brushmenü> – im Zusammenspiel liegen die Tricks

Jede Funktion für sich genommen mag sicher in vielen Zeichnungen die Lösung für bestimmte anstehende Probleme darstellen. Jedoch ist es auch möglich, die Funktionen an sich als Zeichenwerkzeuge zu benutzen oder die Funktionen so nacheinander zu benutzen, da daraus ein ansehnliches Bild wird. Folgendes Beispiel wird das ein wenig erläutern.

– Zeichnen Sie sich mit einem der dickeren eingebauten Pinsel eine gerade, waagrechte Linie in Blau.

– Definieren Sie den eben gezeichneten Strich als einen Brush.

– Wählen Sie aus der Farbpalette Grün aus.

– Aktivieren Sie nun Im <Edge>-Menü <Outline>.

– Wählen Sie aus der Farbpalette Violett aus.

– Aktivieren Sie nun wieder um im <Edge>-Menü <Outline>.

Bisher sind durch diese Aktionen um unseren blauen Strich eine grüne und eine violette Randlinie entstanden. Außerdem werden Sie vielleicht festgestellt haben, daß <Trimm> und <Outline> nur mit selbstdefinierten Brushes benutzt werden kann. Brushes, die Deluxe Paint von vornherein zur Verfügung stellt, können nicht mit <Trimm> und <Outline> bearbeitet werden. Wir haben quasi eine Borte rund um unseren Brush gezaubert. Wiederholen Sie die Prozedur noch ein bis zwei Mal, so daß die Borte deutlicher sichtbar wird.

– Setzen Sie den bisher gestalteten Strich in eine Ecke des Bildschirms.

– Wählen Sie die Funktion gefülltes Viereck an und den kleinsten der eingebauten Pinsel.

– Wählen Sie Braun aus der Palette.

– Zeichnen Sie ein flaches Rechteck, in dem der vorher gezeichnete Strich mit der Borte bequem Platz hat.

So weit, so gut. Nun haben wir einen bunten Strich und ein gelbes Rechteck, für die es keiner besonderen Tricks bedurfte, jetzt aber wird es interessant.

– Definieren Sie den vorhin abgelegten Strich mit der Borte als einen Brush.

– Wählen Sie nun aus dem <Mode>-Kopfleistenmenü die <Color>-Option an. Sie können sie auch einfach durch die F2 Taste aktivieren.

Kaum wurde die <Color>-Option aktiviert, müßte sich Ihr Brush komplett gelb verfärbt haben. Wenn Sie nun irgendeine andere Farbe in der Palette aktivieren, z.B. Schwarz, färbt sich automatisch auch Ihr Brush in derselben Farbe. Sie brauchen allerdings nun keine Angst haben, daß die Arbeit an dem Bortenstrich vergeblich gewesen sei, weil ja nun ausnahmslos alles mit einer anderen Farbe gefüllt ist. Dies liegt ausschließlich daran, daß wir den Colormode aktiviert haben. Wählen Sie im Kopfleistenmenü <Mode> die <Matte>-Option an, wird sich Ihr Brush wieder in den ursprünglichen Originalzustand zurückversetzen.

– Wählen Sie in der Palette Schwarz aus.

– Setzen Sie den schwarzen Strich einmal inmitten des gelben Rechteckes.

– Wählen Sie aus dem <Mode>-Menü die Option <Matte> aus.

– Setzen Sie den Strich mit der Borte, (nachdem Sie <Matte> aktiviert haben, ist dieser wieder als solcher zu sehen) einmal über den schwarzen Fleck auf dem gelben Rechteck. Und zwar so, daß am rechten und am unteren Rand des Bortenstriches noch ein Stück der schwarzen Fläche zu sehen ist.

Sie müssen zugeben, ein wenig sieht das, was wir da gerade eben zusammengesetzt haben, wie eine Fläche aus, über der eine andere Fläche schwebt und einen Schatten wirft. Natürlich lassen sich die einzelnen Komponenten, aus welchen wir unser Bild eben gerade zusammengesetzt haben, noch viel weiter verfeinern und ausarbeiten, was natürlich zusätzlich dessen optische Wirkung verstärkt. Aber wir können noch mehr tricksen. Um wieder einen gemeinsamen Startpunkt zu haben, betätigen Sie jetzt bitte noch einmal <Undo>. Nun sollten Sie wieder Ihre braune Fläche haben ohne den daraufgesetzten Bortenstrich.

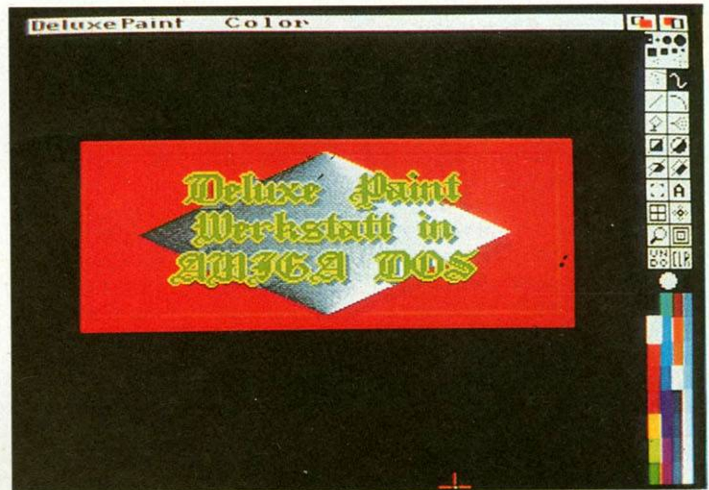
– Öffnen Sie das <Choose Font>-Menü, und wählen Sie die Diamond-Schrifttype in einer Größe von 12 Punkten.

– Aktivieren Sie die Funktion Texteingabe.

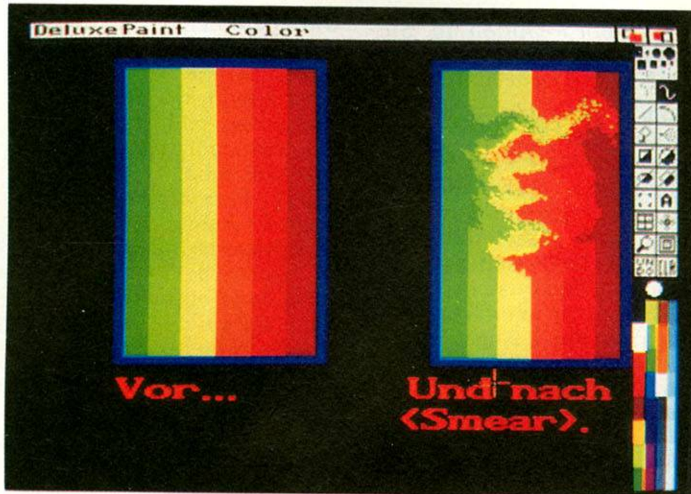
Wenn Sie nun mit Ihrem Mauszeiger in die Zeichenfläche von Deluxe Paint zurückkehren, werden Sie feststellen, daß unter Ihrem Mauszeiger nun ein kleines Rechteck hängt. Dieses Rechteck können Sie nun mit einem Mausklick auf der Zeichenfläche positionieren und danach mit der Texteingabe beginnen. Das Rechteck bestimmt also, an welcher Stelle des Bildschirms der Text beginnt. Sie können natürlich auch 'Backspace' verwenden, um bereits geschriebene Lettern wieder zu entfernen. 'Delete' funktioniert nicht mehr wie gewohnt, sondern wurde mit einem Grafikzeichen neu belegt. Wird 'Return' betätigt, taucht der Cursor unterhalb der Stelle des Bildschirms auf, an der er zum ersten Male postiert wurde. Die Höhenpositionierung erfolgt je nach der Größe der aktuell eingestellten Schrift. Es ist auch möglich, mit den Cursorta-



Mit Deluxe-Paints-Fonts und der Funktion <Outline> ist die Schriftzuggestaltung kein Problem



Ein solches Logo kann unter Verwendung unseres Beispiels und einiger zusätzlicher Handgriffe leicht erstellt werden



Mit <Smear> bearbeitete Farbflächen



Mit <Stencil> angefertigte Farbauszüge aus einem Brush

sten durch den bereits geschriebenen Text zu wandern. Wollen Sie allerdings in bereits Geschriebenes noch etwas einfügen, ist Vorsicht geboten. Deluxe-Paints-Fonts sind als Proportional-Schriften angelegt. Das bedeutet, daß Buchstaben wie ein I weit schmaler sind als ein W. Wenn nun mit den Cursortasten durch den Text gewandert wird, kann es dazu kommen, daß die Corsorposition mit den tatsächlichen Anfängen der Buchstaben nicht mehr übereinstimmen und nachträglich eingegebene Buchstaben völlig deplaziert werden.

– *Wählen Sie die Farbe Gelb aus der Palette.*
– *Positionieren Sie Ihren Textcursor auf dem Zeichenbildschirm, und erstellen Sie ein Wort oder einen Text, der ungefähr so lang ist wie die Bortenlinie, die wir zuvor erstellt haben.*
– *Definieren Sie das eben geschriebene Wort als Brush.*
– *Aktivieren Sie mit der Taste F2 erneut den <Color>-Mode.*
– *Wählen Sie Schwarz aus der Palette.*
– *Setzen Sie dem schwarzen Text einmal auf Ihre Bortenlinie, von der Mitte aus zwei bis drei Pixel nach rechts und unten versetzt.* – *Drücken Sie F1, um in den <Matte>-Mode zurückzukehren.*
– *Setzen Sie Ihr Wort (das nun wieder gelb sein sollte) mitten auf die Bortenlinie.*

Nun sollten rechts und unten unter dem gelben Text der schwarze hervorlugen. Sie sehen, wir haben wieder mit dem Schatteneffekt gearbeitet.

– *Definieren Sie die neue Bortenlinie mit Text als Brush.*
– *Wieder mittels F2 den <Color>-Mode aktivieren und Schwarz aus der Palette wählen.*
– *Setzen Sie wieder einen Schatten, der leicht nach rechts und unten versetzt ist.*
– *Mit F1 wieder in den <Matte>-Mode umschalten und nun über den eben positionierten Schatten, wieder zur Mitte hin versetzt, den eigentlichen Bildausschnitt setzen.*

Im Grunde haben wir auch diesmal nicht viel mehr getan als bei unserem ersten Versuch. Während dieser allerdings so eine Art Minimalversion darstellt, sind bei der zweiten Version schon Text und zusätzliche Effekte verwendet worden. Und trotzdem könnte immer noch mehr gemacht werden. Z.B. könnten einzelne Bildelemente noch gebogen (<Bend>) oder verschoben (<Shear>) werden. Der Schriftzug könnte durch Anwendung der <Outline>-Funktion noch weiter gestaltet werden und vieles andere mehr.
Wie Sie sehen, werden Anwender, die es nicht verstehen, Deluxe-Paint-Funktionen in Wechselwirkung zueinander zu benutzen, nur einen kleinen Teil der echten Möglichkeiten erschließen, die dieses Programm beinhaltet.

Das <Mode>-Menü

Während sich die Funktionen im <Brush>-Menü auf das Arbeiten und Manipulieren von Bildausschnitten beziehen, finden wir im <Mode>-Menü eine ganze Reihe anderer Malwerkzeuge ganz unterschiedlicher Wirkungsweise. Mit zwei dieser Funktionen haben wir in unsereren bisherigen Beispielen schon öfter gearbeitet. Die Rede ist von <Matte> und von <Color>. Alle weiteren Funktionen, bis auf <Smooth>, wurden in den bisherigen Teilen dieses Kurses weitgehend erklärt. <Smooth> jedoch wollen wir uns noch ein wenig genauer ansehen.

– *Öffnen Sie die Palette, und definieren Sie durch Benutzen von <Spread> einen Farbbereich, der 16 Grautöne beinhaltet.*
– *Zeichnen Sie nun ein großes Quadrat, das mit der hellsten Farbe aus Ihren eben definierten 16 Grautönen gefüllt ist.*
– *Klicken Sie die <Airbrush>-Funktion aus dem Basismenü mit der rechten Maustaste an.*
– *Definieren Sie einen großen Radius für die <Airbrush>.*
– *Zeichnen Sie mit <Airbrush> ein nicht zu dichtes Muster mit einer beliebigen Farbe aus den definierten 16 Grautönen.*
– *Wiederholen Sie diesen Vorgang mehrmals. Benutzen Sie dabei immer wieder andere Farben, jedoch immer aus dem Bereich der 16 definierten Grautöne.*

So, die grundsätzlichen Vorbereitungen sind getroffen. Jedoch bliebe da noch ein Wörtchen über den Speicherplatz zu verlieren. An und für sich reserviert sich Deluxe Paint nach dem Start so viel erkennbaren Speicher, wie es belegen kann. Spätestens wenn mit 1 MByte Speicher gearbeitet wird, lädt Ihr Amiga Deluxe Paint komplett in den Speicher. Wer mit der 512-kByte-Version vorliebnehmen muß, muß nach dem Anwählen einer neuen Funktion öfter einmal auf das Diskettenlaufwerk warten, Deluxe Paint lädt dann gerade den Programmteil für die gewünschte Funktion. Neben diesen Effekten macht sich ein magerer Speicher aber auch zu anderen Gelegenheiten bemerkbar, zum Beispiel beim Ausschneiden von Bildausschnitten. Auch die <Smooth>-Funktion hat ihre Tücken. Die Größe des Bildteiles, der mit <Smooth> behandelt werden soll, ist vom Speicher abhängig.

– *Definieren Sie sich einen Brush. Wählen Sie den Bildausschnitt so, daß er aus der Mitte des Rechtecks stammt, das wir mit der <Airbrush>-Funktion eben ein wenig verziert haben. Der Brush sollte in etwa ein Format von 30 x 30 Pixeln haben.*
– *Aktivieren Sie <Smooth> aus dem <Mode>-Menü. Letzte Sicherheit, ob <Smooth> nun aktiv ist oder nicht, gibt ein Blick auf die Kopfleiste. Ist diese nicht durch den Mauszeiger aktiviert und neben dem Deluxe-Paint-Schriftzug ist das Wort <Smooth> zu sehen, ist alles in Ordnung.*
– *Setzen Sie nun Ihren Brush an einer beliebigen Stelle Ihres Vierecks, und betätigen Sie einmal die linke Maustaste.*
– *Warten Sie ab, was passiert.*

Sie werden sehen, daß Deluxe Paint die Kontraste zwischen einzelnen Farben durch das Einfügen passender Farben aus der aktuellen Palette abschwächt. Was vorher klar und gestochen scharf zu erkennen war, sieht nun so aus, als sei es unscharf fotografiert.

Als wir in der Palette die 16 Grautöne definierten, haben wir für den Effekt, den Sie nun sehen, den Grundstein gelegt. Wären in der Palette außer den Farben, die wir im eigentlichen Bild verwendet haben, keine weiteren, differierenden Grautöne vorhanden gewesen außer jenen, die wir schon verwendet haben, hätte <Smooth> das Bild unverändert hinterlassen. Wenn <Smooth> zwei aneinandergrenzende Farben findet, checkt die Funktion erst einmal ab, welche Farbwerte in den beiden Flächen präsent sind, in unserem Beispiel sind dies die Farben Schwarz und Weiß. Da wir ja insgesamt 16 verschiedene Grautöne definierten, liegen logischerweise zwischen dem hellsten (Weiß) und dem dunkelsten Farbton (Schwarz) 14 verschiedene Nuancen. Deluxe Paint setzt nun zwischen Schwarz und Weiß zusätzlich passende (graue) Farbtöne ein, so daß aus dem harten Farbkontrast ein weiches Hinübergleiten in den anderen Farbton erreicht wird. Die ursprüngliche Grundstruktur der Bildformen bleibt davon unberührt. <Smooth>-Effekte können mit <Undo> wieder zurückgenommen werden.

Maskieren wie ein Grafikprofi – <Stencil> grenzenlos!

Wer schon einmal munter drauflos gemalt hat, und dabei, ohne mit der Wimper zu zucken, die Arbeit von Stunden zunichte gemacht hat, wird sich schnell fragen, wie er ganze Bilder oder Bildteile gegen nachträgliches Übermalen schützen kann. Künstler verwenden dafür dünne Folien, die auf ein Bild aufgebracht werden. Aus den Folien werden jene Teile ausgeschnitten, die geschützt werden sollen, und verbleiben auf dem Bild. Natürlich können Sie mit Deluxe Paint keine Folien verwenden, aber das Programm kennt dafür Alternativen: <Stencil> und <Background>.

Mit <Stencil> kann jede einzelne Farbe, gegen das Übermalen mit einer anderen Farbe gesichert werden. Diese Farben können im <Stencil>-Requester sozusagen verriegelt werden. Die Bedienung des <Stencil>-Requesters wurde bereits im zweiten Teil dieses Kurses erklärt. Die andere Funktion, die dem Bildinhalt Schutz verheißt, ist <Background>. Im Grunde wirkt <Background> genauso wie <Stencil>, aber während <Stencil> den Schutz ganz spezifischer Farben zuläßt, schützt Background alle bis dahin erzielten Ergebnisse. Ein kleiner Versuch verdeutlicht dies.

– Löschen Sie den Bildschirm.

– Aktivieren Sie die <Füllen>-Funktion, und füllen Sie den gesamten Zeichenbildschirm mit Grün.

– Wählen Sie nun die Funktion <Freihand Zeichnen>, einen der größten eingebauten Pinsel, und malen Sie mit der Farbe Blau irgendein Muster über die grüne Fläche.

– Aktivieren Sie aus der Menüleiste das <Effects>-Menü und daraus das <Background>-Untermenü.

– Hier aktivieren Sie die Option <Fix>.

– Wählen Sie nun irgendeine andere Farbe, und malen Sie wiederum großflächig über alles bisher Gezeichnete.

– Klicken Sie einmal <CRL> an.

Sie sehen den Effekt selbst. Alles was bis zur Aktivierung von <Fix> gemalt wurde, ist noch vorhanden. Alles was später hinzukam, wurde gelöscht.

Aber neben dem einfachen Schützen können Funktionen aus dem <Effects>-Menü auch anders verwendet werden. Ein kleines Beispiel:

– Zeichnen Sie ein gelbes, gefülltes Quadrat.

– Malen Sie in das gelbe Quadrat einen gefüllten roten Kreis.

– Rufen Sie den <Make Stencil> aus der Kopfleiste auf, und schützen Sie die rote Farbe, aus der auch Ihr Kreis besteht.

Wie in der <Palette> auch müssen die gewünschten Farben nicht unbedingt im Requester angeklickt werden. Es ist durchaus möglich, direkt in das eigentliche Bild hineinzugehen und dort die Regionen anzuklicken, die geschützt werden sollen. Dies ist insbesondere dann sehr hilfreich, wenn mit vielen verschiedenen Farbtönen einer Grundfarbe gearbeitet wird, zum Beispiel 15 verschiedene Grautöne, wie wir Sie in einem vorhergehenden Beispiel benutzt haben. Sollte der Requester sich an einer ungünstigen Stelle des Bildes öffnen, können Sie ihn frei auf der Zeichenfläche umpositionieren. Führen Sie Ihren Mauszeiger in die Kopfleiste des Requesters, und halten Sie dort die linke Maustaste fest. Solange die Maustaste gedrückt bleibt, folgt der Requester und kann neu positioniert werden.

– Ist die rote Farbe gesichert, definieren Sie das Quadrat mit dem Kreis als Brush.

Sie sehen, Erstaunliches passiert. Das gelbe Quadrat ist, wie erwartet, nun der Inhalt unseres Brushes, der rote Kreis hingegen ist verschwunden. Statt seiner klafft ein kreisförmiges, transparentes Loch. Daraus können wir schließen, daß Farben, die mit <Stencil> verriegelt wurden, nicht mehr als Brush definiert werden können. Wir können auf diese Weise auch nur bestimmte Teile aus einem Bildausschnitt herauslösen, indem unerwünschte Farben verriegelt werden. Wird der Brush mit gedrückter, rechter Maustaste definiert, gelingt es sogar, einen Bildteil komplett nach bestimmten Farben zu trennen. Auch hierzu wollen wir ein kleines Beispiel bemühen:

– Zeichnen Sie wieder ein gelbes Rechteck.

– Malen Sie diesmal drei mit unterschiedlichen Farben gefüllte Kreise in das gelbe Rechteck.

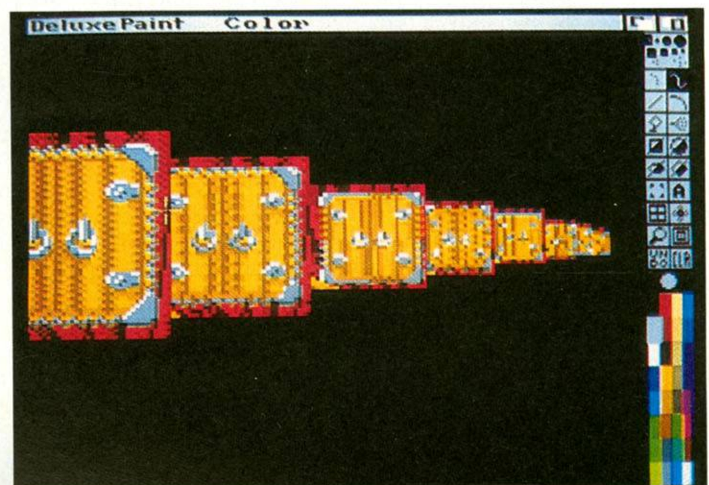
– Öffnen Sie den <Make Stencil>-Requester, und definieren Sie die drei Kreisfarben als <Stencil>.

– Definieren Sie das Rechteck als einen Brush, benutzen Sie diesmal jedoch den rechten Mausknopf.

Als Ergebnis haben wir nun einen gelben, rechteckigen Brush, der drei runde, transparente Löcher aufweist. Die drei Kreise hingegen stehen nun frei, das gelbe Rechteck ist komplett zum Brush geworden. Wird nun der <Stencil> wieder ausgeschaltet, kann der Brush problemlos irgendwo auf dem Arbeitsbildschirm positioniert werden, und der Anwender kann mit den eben freigestellten Bildelementen weiterarbeiten. Es können bis zu 32 verschiedene <Stencils> benutzt werden. Legt der Anwender jedes Ergebnis einer Farbtrennung zwischenzeitlich auf dem Arbeitsbildschirm ab (dabei kann auch der <Spare>-Bildschirm benutzt werden) und definiert dann neue <Stencils>, kann jedes noch so komplizierte Bildteil in alle seine Einzelfarben zerlegt werden. Eine Technik, die bei bedachtem Einsatz ungeahnte Möglichkeiten erschließt.

Perspektive, noch ein Weg die Dinge zu drehen

Der letzte, aber bei weitem nicht der uninteressanteste Punkt des <Effects>-Menüs ist die <Perspective>. Nach-



Ein Brush in verschiedenen Z-Achsen-Positionen dargestellt

dem im letzten Teil dieses Kurses die grundsätzliche Bedienung dieser Funktion erfolgte, wollen wir uns nun ein wenig näher mit der Benutzung der <Perspective> beschäftigen.

Im Grunde genommen fanden alle Operationen, die wir bisher mit Deluxe Paint getätigt haben, wie Malereien auf einem Blatt Papier, also zweidimensional statt. Es war immer eine Fläche, auf der gearbeitet wurde. Selbst Brushes sind nichts anderes als Flächen.

Mit <Perspective> ändert sich dies. Brushes können gedreht, geneigt und gekippt werden, gerade so, als würden sie wirklich auf einem Blatt Papier gemalt sein, das wir nun in den Händen halten und vor den Augen hin und her bewegen und drehen. Ist der gewünschte Drehwinkel erreicht, kann der Anwender Deluxe Paint das Aussehen des Brushes nach der Drehung errechnen lassen und den Brush danach weiterverwenden. Um jedoch zusätzlich Funktionen aus dem <Brush>-Menü benutzen zu können, muß erst <Perspective> ausgeschaltet werden.

Doch auch aus dieser Funktion ist mehr herauszuholen, vorausgesetzt, Sie benutzen Deluxe Paint III. Hier wurden nämlich noch einige kleine Extras eingebaut, die manchen Anwender in Staunen versetzen werden. Zunächst einmal ist es wichtig zu wissen, daß Deluxe Paint III im Speicher des Rechners einen dreidimensionalen Raum simuliert, in dem jeder Brush, sofern <Perspective> eingeschaltet ist, eine relative Position innehat. Dabei stellen Breite und Höhe X- und Y-Achse dar, während die Z-Achse sozusagen die gedachte Linie zwischen den Augen des Anwenders und dem Inneren des Monitors darstellt. Neben einfachem Rotieren und Neigen eines Brushes kann dieser nun auch entlang der Z-Achse bewegt werden. Ein kleines Verfahrensbeispiel:

- *Definieren Sie einen beliebigen Bildausschnitt.*
- *Aktivieren Sie <Do> aus dem <Perspektive>-Untermenü.*
- *Drücken Sie die CTRL-Taste auf der Tastatur, und bewegen Sie die Maus.*
- *Beobachten Sie, was passiert.*

Der Brush wurde stufenlos größer und kleiner, je nachdem, wie die Maus bewegt wurde. Aber es gibt noch einen anderen Weg, einen Brush an der Z-Achse entlangzubewegen. Durch das Drücken von Shift und 'Ö' und Shift Ä kann der Brush schrittweise an der Z-Achse auf und ab bewegt werden.

Wie sicher sind Sie?

Der nächste Teil unserer Deluxe-Paint-Werkstatt wird sich mit einem Thema beschäftigen, das außerordentlich komplex ist, aber leider nur für Benutzer von Deluxe Paint III von Interesse. Es geht um Animation und alles, was damit zusammenhängt. Bis dahin wollen wir es allerdings auch in diesem Monat nicht versäumen, Ihnen die Kontrolle der erarbeiteten Informationen durch ein paar Aufgaben zu ermöglichen.

Aufgabe 1: Erstellen Sie ein Logo unter Verwendung der in dieser Werkstatt erläuterten Techniken.

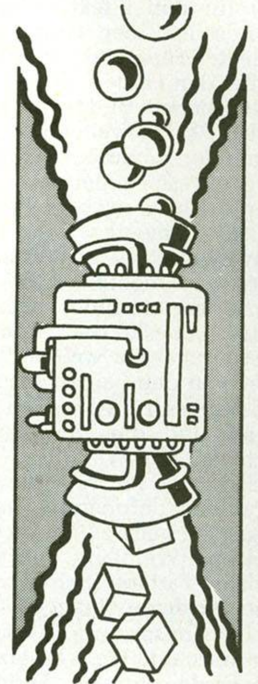
Aufgabe 2: Nehmen Sie einen beliebigen Bildausschnitt, und lösen Sie alle Farben einer Tönung (alle Rottöne oder alle Blautöne) durch die Verwendung der <Stencil>-Farbtrennung heraus.

Aufgabe 3: Erstellen Sie einen beliebigen Schriftzug, und erstellen Sie unter Verwendung der Funktion <Outline> eine verschiedenfarbige Rahmenlinie um den Schriftzug.

Aufgabe 4: Verschieben Sie einen beliebigen Brush an der Z-Achse, und erstellen Sie mehrere verschieden große Abbilder.

Wir wünschen Ihnen noch viel Spaß bei der Arbeit an den Aufgaben und hoffen, Sie auch nächsten Monat wieder zu dieser Werkstatt begrüßen zu dürfen.

(hs)



Haben Sie die richtige Lösung gefunden?

Natürlich werden auch in dieser Folge unseres Kurses die Testfragen des letzten Monats aufgelöst. Ihre Kontrolle, ob eventuell noch ein Blick auf Teil zwei notwendig ist.

Aufgabe 1: Funktion Brush definieren, aktivieren und Bildausschnitt wählen. In der Kopfleiste im Effects-Menü das Untermenü Perspective öffnen und dort Do anwählen. Über den Zehnerblock der Tastatur den Bildausschnitt nach Belieben neigen. Nochmals in das Perspective Untermenü gehen und diesmal Fill Screen aktivieren.

Aufgabe 2: Funktion gefüllte Rechtecke aktivieren und zwei verschiedenfarbige so zeichnen, daß die Flächen aneinandergrenzen. Einen großen Zeichenstift aus dem Sortiment der eingebauten Stifte auswählen. Mit gedrückter linker Maustaste die Maus von einem der beiden Rechtecke in das andere führen.

Aufgabe 3: In der Kopfleiste die Koordinatenanzeige, Cord, einschalten. Mit der Funktion gefülltes Rechteck ein Quadrat zeichnen. An der Koordinatenanzeige kann abgelesen werden, wann die Seiten des Rechteckes gleich lang sind. Das Quadrat als Brush definieren und damit das Schachbrettmuster zeichnen. In der Kopfleiste die Stencil-Funktion aktivieren und im Stencil-Requester die Farbe der Rechtecke verriegeln.

Aufgabe 4: Sobald das gewünschte Bild fertig ist, in der Kopfleiste im Spare-Untermenü die Funktion Copy to Spare aktivieren. Um sich zu vergewissern, daß tatsächlich etwas in den Spare-Bildschirm kopiert wurde, können Sie mit der Taste 'J' zwischen den beiden Arbeitsbildschirmen hin- und herschalten.

Aufgabe 5: Den Bildschirm löschen. Mit der Taste 'J' in den Spare-Bildschirm umschalten. Hier einen Brush definieren. Wieder mit 'J' in den eben gelöschten Bildschirm umschalten.

AMIGA DOS -TIP – die aktuelle Produktanzeige

6600 | 3300

AMIGA MS-DOS

PUBLIC DOMAIN

Leerdisketten 3,5" (2DD) Bulkware m. Aufkleber 100Stück DM 161,00

AMIGA und MS-DOS Preise

Auf 3,5" 2DD Disketten je DM 2,10

Auf 5,25" 2D Disketten je DM 1,10

PD-SOFT Bestellannahme

☎0621-673105 NN-Versand

MacSoft – AMIGA Shop

Public Domain

JEDE DISKETTE AUF 2 DD NUR **4,– DM**

Alle gängigen Serien!
Markendisketten 2 DD!
24-Std.-Versand-Service!
Kostenloser Umtausch defekter Disketten. Keine zusätzliche NN!
Update von Katalogdisketten!
Selber Abholen! NN gespart!
Katalog-Diskette 5,-

Telefon 0231/512603

Mo.-Fr. 16-21 • Samstag 10-16 • BTX MacSoft**

Kasselerstr. 2a • 4600 Dortmund

RUHRSOFT

0234/411958

Erfahrung und Qualität hat ihren Preis

Wir kopieren auf farbigen 2DD-Disketten

EIGENE SERIEN:		Deutsche Serien:	
RPD	- 220	Auge	- 43
Ruhr	- 28	Berlin PD	- 32
Importierte Serien:		Taifun	- 120
		Poseidon	- 400
Fish	- 294	**ACS**	- 191
TBAG	- 33	KICKSTART	- 220
Panorama	- 98	RW	- 17
CC	- 145	RHS	- 107
FAUG	- 85	RMS	- 37
Erotic Bord.*	- 42	KISS	- 131
UKAUG	- 46	**CACTUS**	- 33
S.A.F.E	- 36!!	FRANZ	- 47
Amicus	- 26	ES	- 75

* nur mit Altersnachweis (Super Serie)

3 **DEUTSCHE** Katalogdisketten DM 8,-

Ruhrsoft Scheer - Hängebank 8a - 4630 Bochum 5

Daten- und Organisationssysteme
Hard- und Softwarevertrieb

Ihr Amiga-Fachhändler im Bergischen Land!

Hardware, Software + Zubehör:

512 Kb Erweiterung, A-500	DM 199,–
3,5"-NEC-Laufwerk	DM 239,–
Druckerkabel	DM 17,90
5,25"-TEAC-Laufwerk	DM 299,–

Alle Gigatron-Speichererweiterungen lieferbar
Alle Hurricane + GVP Produkte lieferbar
Diverse Hardware-Erweiterungen vorführbereit

Restbestände bis zu 50% Rabatt

z.B. Stargoose	DM 29,90
Danger Freak	DM 29,90
Trivial Pursuit	DM 39,90
Willow	DM 39,90
Daley Thompson's Olympic	DM 39,90
The Kristal	DM 59,90
Bomb Fusion	DM 29,90
Denaris (wie Katakis)	DM 39,90

u.v.a. Gesamtliste gegen DM 2,– Portoerstattung

Public Domain .... ab DM 4,–

Disketten (mit GARANTIE):

3,5" 2D .... DM 16,90 3,5" 2D farbig .... DM 19,90

Dies ist nur eine kleine Auswahl aus unserem Programm.
Über weitere Produkte (Hard- und Software) können Sie sich gerne in unserem Geschäft informieren. Wie würden uns über Ihren Besuch sehr freuen!

Öffnungszeiten (Büro + Ladengeschäft):
Mo–Fr 10.00–18.30 – Sa 10.00–14.00 – langer Sa 10.00–16.00
Sedanstraße 136 * 5600 Wuppertal 2
Tel. 0202/501500 * Martin Kramer

DONAU-SOFT

24 h-Schnellversand

Neutrale Disketten

3,5" 2DD (100 % errorfree)

	von Sentinel	von SONY/Collossus
bis 99 Stück	1,80 DM	2,10 DM
ab 100 Stück	1,60 DM	1,95 DM
ab 500 Stück	**1,40 DM**	**1,85 DM**

Laufwerke mit allen Extras

3,5" intern	175,– DM
3,5" extern, abschaltbar, Busdurchführung	229,– DM
5,25" extern, wie 3,5" + 40/80-Trackumschaltung	269,– DM

Sim City	97,– DM	GFA-Basic	189,– DM
B.A.D.	77,– DM	GFA-Compiler	97,– DM
Pagestream	369,– DM	512 KB-Erw. (A500)	228,– DM
Zoetrope 1.1	189,– DM	2 MB-Erw. (A500)	698,– DM

Vorkasse: + 5,– DM, Nachnahme: + 8,– DM, Ausland: + 10,– DM

MAIK HAUER

Postfach 1401, 8858 Neuburg Fax: 08431/49800
Tel.: 08431/49798 (bis 22 Uhr) BTX: *Donau-Soft #



Tino Hofstede
Computerservice
An der Windmühle 8
5010 Bergheim 5



WARRIORS, SF-Adventure	19,90
HOTELMASTER, Spiel	19,90
ENGLISCHTRAINER incl. Vokabeln	39,90
FRANZÖSICHTRAINER incl. Vokabeln	39,90
VIDEOVERWALTUNG	49,90
DISK-ETIKETTEN-Druck/Programm	19,90
LIGAVERWALTUNG (6-20 Mannschaften)	49,90
PD-DISKETTEN	ab 2,60
KATALOG (Schutzgebühr in Briefmarken)	1,00

5500 | 3000

AMIGA MS-DOS

PUBLIC DOMAIN

KOPIERT AUF 3,5" UND 5,25" 2DD-DISKETTEN ZU TOP-PREISEN

100 Leerdisketten (3,5"-2DD) in 10er Packs .... DM 170,–

INFO-MATERIAL VON:

A.P.S.-ELECTRONIC

Sonnenborstel 31, 3071 Steimbke

TEL. 05026/1700 FAX 05026/1615

Speichererweiterungen

Amiga 500	- 512 KB intern, abschaltbar	ab	DM	179,--
Amiga 500	- 1 MB intern, abschaltbar, mit Uhr	ab	DM	449,--
Amiga 500	- 1.5 MB intern, abschaltbar, mit Uhr	ab	DM	569,--
Amiga 500	- 2 MB intern, abschaltbar, mit Uhr	ab	DM	689,--
Amiga 1000	- 512 KB intern, abschaltbar	ab	DM	398,--
Amiga 1000	- 2 MB extern, abschaltbar	ab	DM	889,--
Amiga 2000	- 2 MB intern, abschaltbar	ab	DM	998,--
Amiga 2000	- 4 MB intern, abschaltbar	ab	DM	1.698,--
Amiga 2000	- 8 MB intern, abschaltbar	ab	DM	2.998,--

Alle Speichererweiterungen sind autokonfigurierend, abschaltbar und mit sehr schnellen RAM's (100ns und schneller) ausgerüstet!
Durch Megabit-Technologie minimaler Strombedarf.
****12 Monate Garantie****

Floppylaufwerke u. Disketten

Laufwerk 3.5" intern f. Amiga 2000		DM	198,--
Laufwerk 3.5" extern, durchgeschl. Bus, abschaltbar		DM	249,--
Laufwerk 5.25" extern, durchgeschl. Bus, abschaltbar		DM	298,--
3.5" Disketten 2DD 135 tpi 10St. /16,90 50St. /82,--	100 St.	DM	159,--
5.25" Disketten 2D 10 St. / 6,90 50St. / 33,--	100 St.	DM	59,--

B & S Computer-Vertriebs GmbH
Beethovenstr. 33 ; 4172 Straelen 1
Tel: 02834/1249 ; Fax: 02834/6979

Von 9.30 bis 22.00 UHR

AMIGA PD SO GÜNSTIG WIE NOCH NIEMALS ZUVOR
WIR KOPIEREN MIT VERIFY

ÜBER 4500 PD-DISK

R. Dombrowski
Postfach 71 04 62
2000 Hamburg 71

040/ 642 82 25
NEU 24 Std.Versand-Service ohne Aufpreis.

3,5" 2DD	5,25" 2D
PD incl. Qualitätsdisk	PD incl. Qualitätsdisk
1 - 9 à DM 2,80	10 -39 à DM 1,30
10 -79 à DM 2,00	40 -99 à DM 1,10
80- à DM 1,90	100- à DM 1,00
Serienabnahme ab 200 PD à 1,70 DM	Serienabnahme ab 300 PD a' 0,95 DM
WIR HABEN ›80 SERIEN	ALLE UNSERE SERIEN SIND IMMER AKTUELL ›60 !
ABO Möglichkeit!	EIGENE SERIE "ANTARES"

5,25" Markendisk Zuschlag je Staffel von 0,40 DM a' Disk
5,25" Farbdisketten Zuschlag je Staffel von 0,20 DM a' Disk

PD incl. 3,5" 2DD Markendisk.

1 -9 à DM 3,00	10 -79 à DM 2,40
80- à DM 2,30	Sensationelle Neuheiten

ABSOLUT NEU: deutsche Katalogdiskette; auf dieser Diskette ist der Inhalt von 6 normalen Katalogdisketten enthalten incl. ANTARES Menue 4,--DM incl. Porto nur bei Vorkasse (Briefm) Nachnahme Vorkasse (nur Scheck oder Überweisung kein Bargeld +Porto: 6,00DM Nachnahme 8,00 DM incl. Verpackung

Die Arbeit mit komplexen Anwendungsprogrammen wie beispielsweise Deluxe Paint, hat immer wieder Fragen zur Handhabung aufgeworfen, die das Handbuch nicht beantworten kann. Gerade umfangreiche Bedienungsanleitungen werden, aus welchen Gründen auch immer, nur selten komplett und kompetent ins Deutsche übertragen, so daß Anwender, die des Englischen nicht mächtig sind, bei der Recherche im Handbuch noch zusätzlich gehandicapt sind. Genau dieser Personenkreis wird sich über 'Das große Deluxe Paint III Buch' besonders freuen.

Das große Deluxe Paint III Buch

Hier findet der Anwender neben einer detaillierten Erklärung aller dPaint-III-Funktionen auch viele Tips und Tricks zu deren Benutzung.
Wer sich allerdings viel von den angekündigten Maltips und Tricks erhofft, wird ein wenig enttäuscht: Nur selten gehen diese Tips über das rudimentäre Grundwissen hinaus. Besonders wenn es um die Erläuterung der Arbeiten mit zusätzlichen Programmen oder Hardwarepaketen geht, wird der Anwender schlichtweg enttäuscht. So ist zwar ein Kapitel über das Digitalisieren enthalten, dieses ergeht sich jedoch nur oberflächlich über den grundsätzlichen Vorgang und widmet sich dann der Erklärung der Arbeit mit einem speziellen Digitizer.
Nichtsdestotrotz ist 'Das große Deluxe Paint III Buch' eine sinnvolle Ergänzung zu Deluxe Paint, nicht zuletzt, da sich der Inhalt des Buches mit wenigen Ausnahmen auch auf die Arbeit mit Deluxe Paint II anwenden läßt.

(hs/br)

Name: Das große Deluxe Paint III Buch
Autoren: Mirko Langlotz und Nikola Vignjevic
Verlag: DATA Becker
ISBN: 3-89011-369-9
Preis: 39,– DM

BASIC ist seit jeher die Programmiersprache für Anfänger. Nicht allein aus dem Grund, da das AmigaBASIC im Lieferumfang des Grundgerätes enthalten ist, sondern auch weil BASIC eine sehr leicht zu erlernende Sprache ist. Doch BASIC wird von manchen Programmiererkreisen oft verachtet. Sind die Interpreter doch sehr langsam und, wegen der fehlenden Befehle für strukturierte Programmierung, wird sehr oft "Spaghetticode" erzeugt, der die Übersichtlichkeit des Programms stark vermindert.



Dies sollte sich ändern, als auf der CeBit 1986 GFA-BASIC 1.0 für den Atari ST vorgestellt wurde. Was dem ST recht ist, daß kann dem Amiga nur billig sein. GFA-BASIC 3.0 ist auch auf dem Amiga verwirklicht worden. Jeder war sich einig, daß hier ein BASIC-Dialekt erschaffen wurde, der über enorme Befehle verfügt und der mehr an Pascal als an BASIC erinnert.

GFA-BASIC auf dem Amiga

Kurz danach erschienen auch schon die ersten Bücher, die sich dem GFA-BASIC 3.0 widmen. Eines davon heißt GFA-BASIC auf dem Amiga von Heinz Lösch. Es beginnt mit der Installation und den Grundlagen zum Editor. Danach werden Variablentypen und die Befehle, die sich nicht in die übrigen Kapitel integrieren ließen, kurz erläutert. Nachdem ein kurzer Einstieg in die genormte Programmierung (modular und strukturiert) gegeben wurde, wird das Buch auch für den fortgeschrittenen BASIC-Programmierer interessant.

Der Autor hat die Befehle des BASICs in Bereiche wie Grafik, Dateiverwaltung etc. aufgeteilt. Dabei erläutert er zunächst die allgemeinen Grundlagen, geht dann auf die Befehle zu speziellen Themen ein und entwickelt zum Schluß des jeweiligen Kapitels ein Modul für eine Videokassetten-Verwaltung. Dieses Projekt zieht sich wie ein roter Faden durch das Buch und verbindet so die einzelnen Kapitel miteinander. Dieses Programm wird noch einmal am Schluß des Buches zusammengefaßt.

Der Anhang des Buches macht fast die Hälfte des Umfanges aus. Hier werden alle Befehle noch einmal alphabetisch aufgelistet und ausführlich erläutert. Somit kann schnell einmal nachgeschlagen werden, wenn der Programmierer sich über einen Befehl nicht ganz sicher ist.
Insgesamt macht das Buch einen recht guten Eindruck, wenn auch der interessante Inhalt mit nur 190 Seiten etwas kurz ausgefallen ist. Schade, denn im Augenblick fehlt auf dem Buchmarkt ein Werk, das sich intensiver mit der Programmierung in GFA-BASIC befaßt.
Ein Schmankerl, das nicht vergessen werden darf: Auf die Programmierung von Intuition und das Einbinden von Maschinensprache in GFA-BASIC wird besonders ausführlich eingegangen. Damit man die Beispielroutinen nicht aus dem Buch abzutippen braucht, wurden alle Source-Codes auf einer dem Buch beiliegenden Diskette verewigt.

(Robert Marz/br)

Name: GFA-Basic auf dem Amiga
Autor: Heinz Lösch
Verlag: Markt&Technik
ISBN: 3-89090-746-6
Preis: 69,– DM

Das Problem der Bücher, die über Computerspiele berichten, liegt meist in der Aktualität der Spiele. Bis das Buch auf dem Markt erscheint, hat der Zahn der Zeit am Interesse der Spiele genagt. Dieses Manko versucht das Amiga-Spielebuch durch Klassiker, wie Super Huey, Marble Madness oder Defender of the Crown zu überbrücken. Das Buch gliedert sich in vier Kategorien: Simulationen, Strategiespiele, Geschicklichkeitsspiele und Unterhaltungsspiele.

Das große Amiga Spielebuch

Aufgrund der detaillierten Beschreibung der Spiele hilft das Buch über so manche Barriere, die bei diversen Spielen auftreten können, hinweg. Dies zeigt sich besonders bei den Simulationen und den Strategiespielen.
Manch einer, der Probleme mit dem Flugsimulator II hatte, sieht diese durch das erläuterte Menü und durch einige Tips mehr oder weniger gelöst. Bei den Strategiespielen kristallisieren sich besonders Bard's Tale I und II und De-

fender of the Crown heraus. Natürlich werden bei solchen Spielen keine Komplettlösungen abgedruckt. Das Buch regt lediglich durch spannende Vorgeschichten, umfangreiche Tips und Spielvorschläge zu den Programmen an.
Auch die Ballerfreunde wurden nicht vergessen. Der Unterhaltungsteil wartet beispielsweise mit dem Public-Domain-Spiel Amoeba auf. Es handelt sich dabei um eine gelungene Umsetzung des einstigen Spielhallenhits Space-Invaders.
Auch Sierra Onlines midlifecrisis-geplagter Leisure Suit Larry ist im großen Spielebuch vertreten. Hier hilft das Buch mit einschlägigen Tips weiter.
Insgesamt ist 'Das große Amiga Spielebuch' ein gelungenes Nachschlagewerk für Spielfreudige. Durch Farb- bzw. Schwarzweißbilder werden die Texte ausgeschmückt und einige der 35 Spiele auf ihre grafischen Fähigkeiten analysiert.
Hier eine Zusammenstellung der Spiele, die in diesem Buch besprochen werden:

Simulationen
Championship Golf, Ferrari Formula One, Flight Simulator II, Football Manager 2, Gridiron Football, Interceptor, Jet, Leaderboard Golf, Ports of Call, Super Huey

Strategiespiele
Balance of Power, Bard's Tale I und II, Defender of the Crown, Jinxter, Roadwar 2000, Shadowgate, Uninvited

Geschicklichkeitsspiele
Amoeba, Barbarian, Crystal Hammer, Emerald Mine, Fire Power, Garrison, Jinks, Marble Madness, Obliterator, Plutos, Test Drive

Unterhaltungsspiele
ChessMaster 2000, Leisure Suit Larry, Shanghai

(Jürgen Seibel/br)

Name: Das große Amiga Spielebuch
Autoren: A. Schmidt, J. Hertwig
Verlag: technicSupport
ISBN: 3-926847-02-6
Preis: 49,– DM

In Sachen Grafik gehört der Amiga zu der Elite der Computer. Die Grafiken finden dabei in vielen Bereichen Anwendung, angefangen von Spielen für den Heimbereich über Animationen in der Unterhaltungsindustrie, bis hin zu technischen Zeichnungen in wissenschaftlichen Betrieben. Doch zurück zu den privaten Bereichen:
Wer sich mit dem Genre Grafik näher befaßt, wird immer wieder auf das Wort Genlock-Interface stoßen. Mit einem solchen Gerät kann man den Computer mit einem Videorecorder verbinden. Das Genlock-Interface verbindet nun Amigagrafiken mit den Bildern des Videorecorders. Dabei wird die Hintergrundfarbe der Grafiken ausgeblendet, damit diese nicht die Videobilder überlagert. Die Anwendung können beispielsweise bei Videountertiteln oder Filmvorspännen sein.



Erfolgreich arbeiten mit Video und Computer

Mit diesem Thema beschäftigt sich ein Buch aus dem Verlag Gabriele Lechner. Das Buch läßt sich grob in zwei Teile unterteilen. Im ersten Teil wird die benötigte Hard- und Software vorgestellt. Nachdem die Vor- und Nachteile der verschiedenen Videosysteme, Zubehör, wie Audio-Videomischer, Malprogramme, Digitizer und Genlock-Interfaces ausführlich vorgestellt wurden, folgt jeweils eine Art Einkaufsführer, der sich mit einzelnen Produkten auseinandersetzt.
Besonders ausführlich werden hier die Videosysteme behandelt: Extras, die vorhanden sein sollten, Bandkosten pro Minute und ähnliches mehr.
Die Einkaufsführer sind noch einmal in einzelne Bereiche unterteilt, so daß die Übersichtlichkeit gewahrt bleibt. Im letzten Kapitel geht es dann ans Eingemachte: ein Video-Praxis-Workshop ist angesagt. Hier wird an praktischen Beispielen das Verwirklichen von Projekten geübt. So wird gezeigt, wie man einen Titel für ein bestimmtes Filmthema aufmacht. Als Beispiele seien hier 'Urlaub in Frankreich', 'Ein Besuch im Zoo', oder 'Sportschau' angeführt. Diesem Workshop schließt sich ein Intensivkurs zu 3-D-Videoeffekten und Animationen an. Der Leser lernt dabei an einem Beispiel, einen Wirbeltitel mit dem Flip-Modul zu erstellen. Ein Wirbeleffekt kann beispielsweise das Drehen und Verformen von Bildausschnitten, wie sie aus dem Fernsehen bekannt sind, sein.
Das Buch 'Erfolgreich arbeiten mit Video und Computer' ist ein empfehlenswertes Buch, will man sich mit dem Thema Grafik und Video genauer befassen. Die Einkaufsführer, die einen beträchtlichen Teil des Buches darstellen, sind eine große Hilfe für Einsteiger, die sich auf dem Markt noch nicht so gut auskennen, und auch die Workshops bringen einen optimalen Lerneffekt. Für Videofans, die einen Amiga besitzen (und über das nötige Kleingeld für die nicht ganz billigen Zusatzgeräte verfügen), ist dieses Buch guten Gewissens zu empfehlen.

(Robert Marz/br)

Name: Erfolgreich arbeiten mit Video und Computer
Autor: Volker Schmidtmann
Verlag: Gabriele Lechner
ISBN: 3-926858-14-1
Preis: 69,– DM

Stellen Sie sich vor, Sie sitzen an Ihrem Amiga und versuchen verzweifelt, ein Musikstück zu komponieren. Plötzlich macht es "Puff" – und hinter Ihnen steht der geniale Komponist Wolfgang Amadeus Mozart. Wären Sie nicht genauso verblüfft wie Eugen B., die Hauptperson im Musikbuch aus dem Hause DATA Becker?

Das Musikbuch

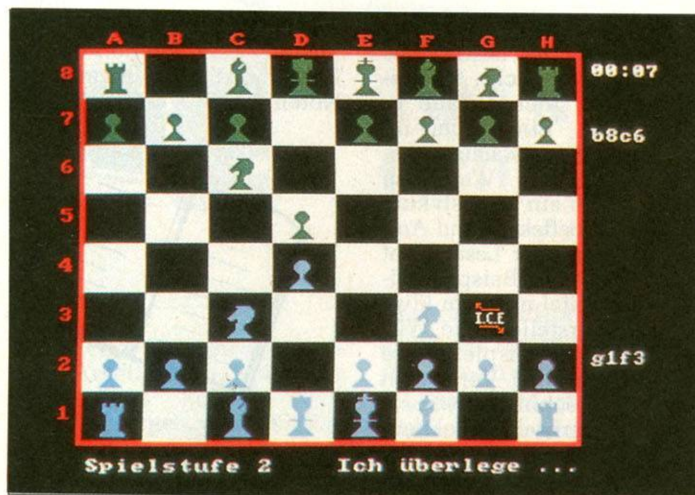
Da zu Amadeus' Zeiten die Computer noch nicht erfunden waren, erklärt Eugen B. die Handhabung seines Musikprogramms Sonix von der Firma Aegis.
Leider hat Eugen B. keinen blassen Dunst, wie man an das Projekt "Komponieren von Musikstücken" herangeht. Klar, daß Mozart diesen Part übernehmen muß. In lockerer Form bringt Musikus Mozart dem Helden dieses Buchs jetzt den Umgang mit Noten bei.



"Das Musikbuch" widmet sich all denen, die gerne auf dem Amiga Musik machen. Dabei wird hauptsächlich auf das Musikprogramm Sonix eingegangen, aber auch andere Musikeditoren (Deluxe Music Construction Set, Aegis Audiomaster, Instant Music und Dynamic Drums) werden besprochen.
Im Sonix-Teil wird anfangs das Verändern von Hüllkurven und das Einladen verschiedener Instrumente besprochen. Da Mozart fasziniert von James-Bond-Filmen ist, setzt er sich daran, die Titelmelodie aufs Papier bzw. auf den Amiga umzusetzen. Diese Komposition zieht sich durch das gesamte Buch und wird Stück für Stück ausgebaut, bis daraus schließlich ein vierstimmiges Lied entsteht.
Ein Teil des Buches bezieht sich auf harmonische Begleitungen, den Dreiklängen sowie den Notenfunktionen mit leicht verständlichen Beispielen.
Doch auch das Thema Digitalisieren von Soundsamples kommt nicht zu kurz: Anhand des Programms Record Maker für den Deluxe Sound Digitizer werden nützliche Tips und Tricks gegeben. Referenzkarten für die bereits erwähnten Musikprogramme bilden den Abschluß dieses Buches.
Allein schon der romanhafte Stil, mit dem der Leser konfrontiert wird, ist es wert, dieses Buch zu lesen. Doch auch für diejenigen Musiker, die Schwierigkeiten mit dem Notenlesen haben, ist das Musikbuch ein schönes Lehrbuch.

(br)

Name: Das Musikbuch
Autoren: Christian Spanik, Thomas Tai, Holger Hahn
Verlag: DATA Becker
ISBN: 3-89011-215-3
Preis: 49,– DM



Niklas Nebel, Michael Cordes

Schach dem Amiga

Königs Bauer ziehe an...

Schach, ein aus dem Orient stammendes Brettspiel, ist seit dem 10. Jahrhundert n. Chr. in Europa bekannt. Anfänglich noch in der oberen Gesellschaftsschicht verbreitet, sammelten sich im Laufe der Zeit immer mehr Schachfreunde. Schach-Clubs wurden gegründet, und mittlerweile übernehmen sogar speziell für diesen Denksport konzipierte Computer den Part des Gegners. Leider sind solche Schachcomputer noch ziemlich teuer. Daß dies nicht sein muß, beweist unser Programm des Monats.

Das Programm ist mit dem Devpac-Assembler geschrieben und umfaßt in der assemblierten Form knapp neun kByte. Die Steuerung erfolgt komplett mit der Maus, Benutzerfreundlichkeit ist somit angesagt.

Die Spielfiguren können mit jedem IFF-fähigen Grafikprogramm selbst erstellt werden: Speichern Sie die Grafik dann als Datei *fig* auf der Schachdiskette ab. Dabei ist es egal, ob sie als Brush oder als Picture abgespeichert wird, da unser Schachprogramm beide Formate einladen und verarbeiten kann. Es können gleichzeitig 32 Farben benutzt werden, also fünf Bitplanes bei 320 Punkten pro Zeile. Der Phantasie sind somit kaum Grenzen gesetzt. Achten Sie bitte bei der Erstellung der Spielfiguren auf deren Größe von 32x24 (Breite x Höhe) Pixel. Das Hauptprogramm lädt die Figurendatei dann selbständig nach. Bevor Sie also eine Partie spielen wollen, sollten Sie sich solch eine fig-Datei erstellen.

... Läufer, Dame folgen dann

Drei Spielstärken hat das Programm zu bieten: von der Anfängerstufe bis zur Partie für Profis, bei der die Rechenzeit bis zu 20 Minuten betragen kann. Voreingestellt ist die Spielstufe zwei. Weiterhin ist es möglich, während des Spiels einen Seitenwechsel vorzunehmen (ist dann von Nutzen, wenn man nicht gern verliert...) oder Spielstände auf Diskette abzuspeichern bzw. einzuladen.

Niklas Nebel und Michael Cordes haben es als erste geschafft, ihr eingesandtes Listing als Programm des Monats in der AMIGA DOS wiederzufinden. Dies soll Ihnen, liebe Leser, als Anreiz dienen, es auch einmal zu versuchen.

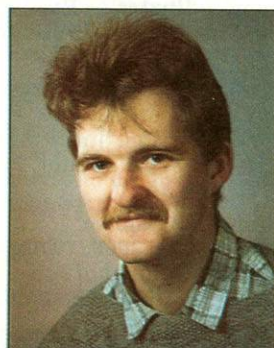
Das Verfahren ist recht einfach: Unter allen Programmeinsendungen sucht sich die Redaktion das Programm aus, was den Lesern der AMIGA DOS wohl am meisten gefallen könnte, und dies wird dann mit 1500 DM prämiert. Dabei ist es unerheblich, in welcher Sprache es programmiert wurde, Hauptsache ist, daß es pfiffig und übersichtlich geschrieben wurde und den Lesern viel Freude bringt. Also wie wär's? Beteiligen Sie sich – vielleicht ist **IHR** Programm demnächst das 'Programm des Monats'!

Spannende Schachpartien wünscht Ihnen Ihr AMIGA-DOS-Team.

(br)



SCHACH
von
Niklas Nebel
und
Michael Cordes
Personenbeschreibung



Name: Nebel
Vorname: Niklas
geb.: 08.06.1969
Beruf: Informatik-Student
Hobbys: Programmieren, Lesen

Die Anfänge machte ein VC20, den bald ein C-64 ablöste. Dabei wurde Wert auf die Programmiersprachen BASIC und Assembler gelegt. Es folgte ein Atari, der zusätzlich in C programmiert wurde. Mittlerweile ist ein Amiga hinzugekommen, der hauptsächlich in Assembler programmiert wird.

Name: Cordes
Vorname: Michael
geb.: 04.10.1967
Beruf: Informatik-Student
Hobbys: Programmieren, Tennis

Der erste Computer war ein C-64. Es wurde eigentlich so alles programmiert, was es an Programmiersprachen auf diesem Homecomputer gab, allerdings hat Assembler den größten Teil ausgemacht. Beim Amiga liegt sein Hauptaugenmerk ebenfalls auf Assembler.

Listings

```
OpenLib         equ     -408
CloseLib        equ     -414
OpenScreen      equ     -198
CloseScreen     equ      -66
OpenWindow      equ     -204
CloseWindow     equ      -72
AllocMem        equ     -198
FreeMem         equ     -210
Read            equ      -42
Write           equ      -48
Output          equ      -60
Open            equ      -30
Close           equ      -36
LoadRGB4        equ     -192
RectFill        equ     -306
SetAPen         equ     -342
SetBPen         equ     -348
PolyDraw        equ     -336
Move            equ     -240
Text            equ      -60
SetMenuStrip    equ     -264
Wait            equ     -318
GetMsg          equ     -372
ReplyMsg        equ     -378
AddIntServer    equ     -168
RemIntServer    equ     -174
AddGadget       equ      -42
RefreshGadgets  equ     -222
RemoveGadget    equ     -228
ActivateGadget  equ     -462

intreq          equ      $9c
intreqr         equ      $1e

rowl = 152
rowr = 153
rosl = 154
rosr = 155
mat  = 156
epw  = 158
eps  = 159

main    move.l  a7,stackp

        bsr     init
        bsr     iniscrn
        bsr     loadpic
        bsr     getfont
        bsr     brett
        bsr     intini

start   bsr     inifld
        bsr     setup
        bsr     getplay
        moveq   #1,d7
        bsr     setlev

mloop   clr     vbllock
        lea     ztxt1(pc),a0
        bsr.s   ztxt
        bsr     clrtime
        bsr     getzug
        not.b   spieler
wechsel lea     ztxt2(pc),a0
        bsr.s   ztxt
        bsr     clrtime
        bsr     compi
*       bsr.s   beep
        not.b   spieler
        bra.s   mloop

*       btst    #7,$bfe001
*       bne.s   mloop

error   bsr     freeint
        bsr     clrscrn
        bsr     clear

        moveq   #0,d0
        rts

ztxt    move    #146,d0
        move    #240,d1
        moveq   #16,d2
        bsr     prtxt
        rts

ztxt1   dc.b    "Du bist am Zug !"
ztxt2   dc.b    "Ich überlege ..."

setlev  addq    #1,d7
        move    d7,ltiefe
        or      #"0",d7
        move.b  d7,.nr

        moveq   #18,d0
        move    #240,d1
        moveq   #16,d2
        lea     .txt(pc),a0
        bsr     prtxt
```

Listing SCHACH

```
        rts

.txt    dc.b    "Spielstufe "
.nr     dc.b    "     "

change  move.l  feldbuf(pc),a5
        move.l  stackp(pc),a7
        bra     wechsel

neustrt move.l  stackp(pc),a7
        bra     start

abbruch move.l  stackp(pc),a7
        bra     error

weiter  move.l  stackp(pc),a7
        bra     mloop

getfont moveq   #0,d0
        moveq   #6,d1
        moveq   #11,d2
        lea     .string(pc),a0
        bsr     prtxt

        move.l  font(pc),a1
        move.l  plane0(pc),a0
        moveq   #10,d7
.11     moveq   #7,d6
.12     move.b  (a0),(a1)+
        clr.b   (a0)
        lea     40(a0),a0
        dbf     d6,.12
        lea     -319(a0),a0
        dbf     d7,.11

        rts

.string dc.b    "0123456789:",0

clrtime move    #1,count
        move    #-1,sek
        clr     min
        rts

intini  move.l  4.w,a6
        moveq   #5,d0
        lea     intstr(pc),a1
        jsr     AddIntServer(a6)
        rts

freeint move.l  4.w,a6
        moveq   #5,d0
        lea     intstr(pc),a1
        jsr     RemIntServer(a6)
        rts

vbl     tst     vbllock
        bne     .11
        subq    #1,count
        bne     .11
        move    #50,count
        addq    #1,sek
        cmp     #60,sek
        bne.s   .anz
        clr     sek
addq    #1,min
.anz    movem.l d2-d7/a0/a2-a4,-(a7)
        lea     ztext(pc),a0

        move    min(pc),d0
        ext.l   d0
        divu    #10,d0
        move.b  d0,(a0)+
        swap    d0
        move.b  d0,(a0)+
        addq.l  #1,a0

        move    sek(pc),d0
        ext.l   d0
        divu    #10,d0
        move.b  d0,(a0)+
        swap    d0
        move.b  d0,(a0)+

        lea     ztext(pc),a0
        moveq   #4,d7
        move.l  font(pc),a2
        move.l  plane0(pc),a1
        lea     835(a1),a1
.12     move.b  (a0)+,d0
        ext.w   d0
        lsl     #3,d0
        lea     (a2,d0.w),a3
        moveq   #7,d6
.13     move.b  (a3)+,(a1)
        lea     40(a1),a1
        dbf     d6,.13
        lea     -319(a1),a1
        dbf     d7,.12

        movem.l (a7)+,d2-d7/a0/a2-a4
.11     rts

ztext   dc.b    0,0,10,0,0
```

```
prtxt   move.l  GfxBase(pc),a6

        movem.l a0/d2,-(a7)
        movem.l d0-d1,-(a7)

        move.l  rp(pc),a1
        moveq   #1,d0
        jsr     SetAPen(a6)

        move.l  rp(pc),a1
        movem.l (a7)+,d0-d1
        jsr     Move(a6)

        move.l  rp(pc),a1
        movem.l (a7)+,a0/d0
        jsr     Text(a6)
        rts

getplay move.b  #0,spieler
        rts

menuoff move.l  window(pc),a0
        move.l  #$40000,82(a0)
        rts

menuon  move.l  window(pc),a0
        move.l  #$40108,82(a0)
        rts

compi   bsr.s   menuoff
        move    ltiefe(pc),tiefe
        move.b  spieler(pc),bspiel
        bsr.s   .czug
        move.b  bspiel(pc),spieler
        bsr.s   menuon

        mcve    d2,d3
        beq     gamover
        lsr     #8,d3
        move.b  spieler(pc),d0
        bsr     zug

        moveq   #0,d0
        move.b  d2,d0
        divu    #10,d0
        move    d0,d1
        swap    d0
        subq    #1,d0
        subq    #2,d1
        ext.l   d3
        divu    #10,d3
        move.l  d3,d2
        swap    d2
        subq    #1,d2
        subq    #2,d3
        bsr     bzug

        rts

.lx     movem   (a7)+,d0-d2
        bra     .l2

.czug   move.l  zliste(pc),a0
        move.b  spieler(pc),d0
        bsr     mliste
        beq     .nomove
        subq    #1,d0
        move.l  zliste(pc),a1
        move    #-32768,d1
        tst.b   spieler
        beq.s   .cw
        move    #32767,d1
.cw     moveq   #0,d2
.l1     movem   d0-d2,-(a7)
        bsr     nbrett
        move.b  (a1)+,d3
        move.b  (a1)+,d2
        beq     .lx
        move.b  spieler(pc),d0
        bsr     zug

        move    tiefe(pc),d3              *tst*
        bne.s   .lrek                     rekursiv aufrufen

        move    mat(a5),d3                Wert

        tst.b   54(a5)
        beq.s   .n1
        bmi.s   .n2
        addq    #8,d3
        bra.s   .n1
.n2     subq    #8,d3
.n1     tst.b   55(a5)
        beq.s   .n3
        bmi.s   .n4
        addq    #8,d3
        bra.s   .n3
.n4     subq    #8,d3
.n3     tst.b   64(a5)
        beq.s   .n5
        bmi.s   .n6
        addq    #8,d3
        bra.s   .n5
.n6     subq    #8,d3
.n5     tst.b   65(a5)
        beq.s   .n7
        bmi.s   .n8
        addq    #8,d3
        bra.s   .n7
.n8     subq    #8,d3
.n7
        move.b  $bfd800,d0                Zufall
        and     #7,d0
        add     d0,d3

.lc     movem   (a7)+,d0-d2
        tst.b   spieler
        bne.s   .l3
        cmp     d3,d1
        bge.s   .l2
        move    d3,d1
        move    -2(a1),d2                 "bester" Zug
        bra.s   .l2
.l3     cmp     d3,d1
        ble.s   .l2
        move    d3,d1
        move    -2(a1),d2                 "bester" Zug
.l2     lea     -160(a5),a5
        dbf     d0,.l1
        rts

.lrek   move.l  a1,-(a7)
        not.b   spieler
        add.l   #1024,zliste
        subq    #1,tiefe
        bsr     .czug
        sub.l   #1024,zliste
        addq    #1,tiefe
        not.b   spieler
        move    d1,d3                     Wert
        move.l  (a7)+,a1
        bra.s   .lc

.nomove moveq   #0,d2
        tst.b   spieler
        bne.s   .cl2
        move    #-30000,d1
        rts
.cl2    move    #30000,d1
        rts

gamover move.l  feldbuf(pc),a5
        moveq   #0,d0
        bsr     chtest
        beq.s   .go2
        lea     wlose(pc),a0
        bra.s   .gox
.go2    moveq   #-1,d0
        bsr     chtest
        beq.s   .go3
        lea     slose(pc),a0
        bra.s   .gox
.go3    lea     remis(pc),a0

.gox    moveq   #32,d2
        moveq   #18,d0
        move    #240,d1
        bsr     prtxt

.gol    move    #1,vbllock
        bsr     readmen
        bmi.s   .gol

        tst     d5
        bne.s   .goml
        tst     d6
        beq.s   .gol
        bra     abbruch

.goml   cmp     #2,d5
        bne.s   .gom2
        cmp     #2,d6
        bne.s   .gol
        bra     neustrt

.gom2   tst     d6
        beq     load
        bsr     save
        bra.s   .gol


wlose   dc.b    " Spielende !  Schwarz gewinnt ! "
slose   dc.b    " Spielende !  Weiß gewinnt !    "
remis   dc.b    " Spielende !  Remis !           "

wait    btst    #6,$bfe001
        bne.s   wait
        rts

inifld  lea     feld(pc),a5
        lea     anf(pc),a0
        moveq   #21,d1
        moveq   #1,d2
        moveq   #-127,d5
        moveq   #7,d3
```

Listing SCHACH

```
        moveq   #-128,d6
.l1     move.b  (a0),(a5,d1.w)
        move.b  (a0)+,d4
        or.b    d6,d4
        move.b  d4,70(a5,d1.w)
        move.b  d2,10(a5,d1.w)
        move.b  d5,60(a5,d1.w)
        addq    #1,d1
        dbf     d3,.l1

        lea     20(a5),a0
        move.l  feldbuf(pc),a5
        lea     20(a5),a1
        moveq   #19,d0
.l2     move.l  (a0)+,(a1)+
        dbf     d0,.l2

        move.b  #25,120(a5)           Könige
        move.b  #95,136(a5)
        move.l  #-1,rowl(a5)          Rochade
        clr.l   mat(a5)               Material=0, ke
in en passant

        rts

feld    dc.b    -1,-1,-1,-1,-1,-1,-1,-1,-1,-1
        dc.b    -1,-1,-1,-1,-1,-1,-1,-1,-1,-1
        dc.b    -1,0,0,0,0,0,0,0,0,-1
        dc.b    -1,0,0,0,0,0,0,0,0,-1
        dc.b    -1,0,0,0,0,0,0,0,0,-1
        dc.b    -1,0,0,0,0,0,0,0,0,-1
        dc.b    -1,0,0,0,0,0,0,0,0,-1
        dc.b    -1,0,0,0,0,0,0,0,0,-1
        dc.b    -1,0,0,0,0,0,0,0,0,-1
        dc.b    -1,0,0,0,0,0,0,0,0,-1
        dc.b    -1,-1,-1,-1,-1,-1,-1,-1,-1,-1
        dc.b    -1,-1,-1,-1,-1,-1,-1,-1,-1,-1

anf     dc.b    4,2,3,5,6,3,2,4

getzug  move.l  zliste(pc),a0
        move.b  spieler(pc),d0
        bsr     mliste                mögliche Züge
berechnen
        beq     gamover               keine Züge ?
        move    d0,.zuege             Anzahl

.get    bsr     getm
        move    d1,d2
        mulu    #10,d2
        add     d0,d2
        add     #21,d2

        move.l  zliste(pc),a0
        move    .zuege(pc),d7
        subq    #1,d7
.ch1    move.b  (a0)+,d3
        cmp.b   d3,d2                 Zug für Figur?
        beq.s   .ch2
        addq.l  #1,a0
        dbf     d7,.ch1
        bra.s   .get                  kein Zug mögli
ch

.ch2    move.b  d2,.von               Feldnummer spe
ichern
        movem   d0-d1,-(a7)
        bsr     mark                  Feld markieren

        bsr     waitm                 warten bis Mau
s losgelassen

.get2   bsr     getm
        move    d1,d2
        mulu    #10,d2
        add     d0,d2
        add     #21,d2

        cmp.b   .von,d2               gleiches Feld?
        beq.s   .undo

        or      .von,d2               "von" ins High
-Byte
        move.l  zliste(pc),a0
        move    .zuege(pc),d7
        subq    #1,d7
.cs1    move    (a0)+,d3
        cmp     d3,d2                 dieser Zug mög
lich?
        beq.s   .cs2
        dbf     d7,.cs1
        bra.s   .get2                 kein Zug mögli
ch

.cs2    move    d2,d3
        lsr     #8,d3                 "nach"-Byte lö
schen
        ext.w   d2                    "von"-Byte lös
chen

        move    d0,-(a7)
        move.b  spieler(pc),d0
```

Listing SCHACH

```
        bsr.s   zug
        move    (a7)+,d0

        movem   (a7)+,d2-d3
        bsr     bzug

        rts

.undo   movem   (a7)+,d0-d1
        bsr     show                  wieder normal
darstellen
        bsr     waitm
        bra     .get                  neu auswählen

.zuege  ds.w    1
.von    dc.w    0

zug     ext.w   d2
        ext.w   d3

        lea     (a5,d3.w),a0

        move.b  (a0),d6
        cmp.b   #6,d6
        bne.s   .zl1
        move.b  d2,120(a5)
        bra.s   .zlx
.zl1    cmp.b   #134,d6
        bne.s   .zl2
        move.b  d2,136(a5)
.zl2
.zlx
        clr     epflag                epflag + roflag

        tst.b   d0
        bne.s   .z2
        clr.b   epw(a5)

        cmp.b   #1,(a0)
        bne     .z1
        move.b  d2,d7
        sub.b   d3,d7
        cmp.b   #20,d7
        bne.s   .w1
        moveq   #0,d7
        move.b  d2,d7
        lea     xbittab(pc),a2
        move.b  -20(a2,d7.w),epw(a5)   ep erlauben
        bra.s   .z1

.w1     cmp.b   #11,d7
        beq.s   .w2
        cmp.b   #9,d7
        bne.s   .z1
.w2     tst.b   (a5,d2)
        bne.s   .z1
        clr.b   -10(a5,d2)
        move.b  #1,epflag
        addq    #1,mat(a5)
        bra.s   .z1

.z2     clr.b   eps(a5)

        cmp.b   #129,(a0)
        bne.s   .z1
        move.b  d3,d7
        sub.b   d2,d7
        cmp.b   #20,d7
        bne.s   .s1
        moveq   #0,d7
        move.b  d2,d7
        lea     xbittab(pc),a2
        move.b  -20(a2,d7.w),eps(a5)
        bra.s   .z1

.s1     cmp.b   #11,d7
        beq.s   .s2
        cmp.b   #9,d7
        bne.s   .z1
.s2     tst.b   (a5,d2)
        bne.s   .z1
        clr.b   10(a5,d2)
        move.b  #1,epflag
        subq    #1,mat(a5)

.z1     move.b  (a0),d7
        and     #7,d7
        cmp     #6,d7
        bne.s   .z3
        move.b  d2,d7
        sub.b   d3,d7
        cmp.b   #-2,d7
        bne.s   .zr1
        move.b  -4(a0),-1(a0)
        clr.b   -4(a0)
        move.b  #-1,roflag
        bra.s   .zk

.zr1    cmp.b   #2,d7
        bne.s   .zk
        move.b  3(a0),1(a0)
        clr.b   3(a0)
        move.b  #1,roflag
```

```
        bra.s   .zk
.z3     cmp     #4,d7
        bne.s   .z5
        cmp.b   #21,d3
        bne.s   .nr1
        clr.b   rowl(a5)
        bra.s   .z5
.nr1    cmp.b   #28,d3
        bne.s   .nr2
        clr.b   rowr(a5)
        bra.s   .z5
.nr2    cmp.b   #91,d3
        bne.s   .nr3
        clr.b   rosl(a5)
        bra.s   .z5
.nr3    cmp.b   #98,d3
        bne.s   .z5
        clr.b   rosr(a5)
        bra.s   .z5
.zk     cmp.b   #25,d3
        bne.s   .nr4
        clr     rowl(a5)
        bra.s   .z5
.nr4    cmp.b   #95,d3
        bne.s   .z5
        clr     rosl(a5)
.z5     cmp.b   #1,(a0)
        bne.s   .z6
        cmp.b   #91,d2
        blt.s   .z7
        move.b  #5,(a5,d2.w)
        addq    #8,mat(a5)
        bra.s   .zu
.z6     cmp.b   #129,(a0)
        bne.s   .z7
        cmp.b   #28,d2
        bgt.s   .z7
        move.b  #133,(a5,d2.w)
        subq    #8,mat(a5)
        bra.s   .zu
.z7     moveq   #0,d7
        move.b  (a5,d2.w),d7
        beq.s   .z8
        bmi.s   .wz1                    weiß schlägt s
chwarz
        lea     werte(pc),a2
        add     d7,d7
        move    -2(a2,d7.w),d7
        sub     d7,mat(a5)
        bra.s   .z8
.wz1    lea     werte-256(pc),a2
        add     d7,d7
        move    -2(a2,d7.w),d7
        add     d7,mat(a5)
.z8     move.b  (a0),(a5,d2.w)          kopieren
.zu     clr.b   (a0)
        rts
bzug    move    #1000,count
        lea     .text(pc),a0
        move    d2,d4
        add     #"a",d4
        move.b  d4,(a0)+
        move    d3,d4
        add     #"1",d4
        move.b  d4,(a0)+
        move    d0,d4
        add     #"a",d4
        move.b  d4,(a0)+
        move    d1,d4
        add     #"1",d4
        move.b  d4,(a0)+
        tst.b   epflag
        beq.s   .bz
        cmp     #4,d3
        bne.s   .bzs
        movem.l d0-d7/a0-a6,-(a7)
        subq    #1,d1
        bsr     kill
        movem.l (a7)+,d0-d7/a0-a6
        bra.s   .bz
.bzs    movem.l d0-d7/a0-a6,-(a7)
        addq    #1,d1
        bsr     kill
        movem.l (a7)+,d0-d7/a0-a6
.bz     tst.b   roflag
        beq.s   .bz1
        bmi.s   .br1
```

Listing SCHACH

```
        movem   d0-d3,-(a7)
        movem   d0-d1,-(a7)
        addq    #1,d0
        bsr     kill
        movem   (a7)+,d0-d1
        subq    #1,d0
        bsr     show
        movem   (a7)+,d0-d3
        bra     .bz1
.br1    movem   d0-d3,-(a7)             lange Rochade
        movem   d0-d1,-(a7)
        subq    #2,d0
        bsr     kill
        movem   (a7)+,d0-d1
        addq    #1,d0
        bsr     show
        movem   (a7)+,d0-d3
.bz1    movem   d2-d3,-(a7)
        bsr     show
        movem   (a7)+,d0-d1
        bsr     kill
        moveq   #63,d1
        tst.b   spieler
        bne.s   .zt1
        move    #182,d1
.zt1    move    #280,d0
        moveq   #4,d2
        lea     .text(pc),a0
        bsr     prtxt
        rts
.text   ds.l    1
mliste  move.l  a0,-(a7)
        moveq   #21,d1
        moveq   #79,d7
.loop   move.b  (a5,d1.w),d2
        beq     .weiter
        bmi.s   .c1
        tst.b   d0                      Farbe am Zug
        bne     .weiter
        lea     eps(a5),a4
        bra.s   .zug
.c1     tst.b   d0
        beq     .weiter
        cmp.b   #255,d2                 Rand ?
        beq     .weiter
        lea     epw(a5),a4
.zug    lea     (a5,d1.w),a2            momentanes Feld
        and     #7,d2
        add     d2,d2
        add     d2,d2
        lea     .jmptab(pc),a1
        move.l  -4(a1,d2.w),a1
        jmp     (a1)
.zb     tst.b   d0
        bne.s   .b1
        moveq   #10,d2                  weiss -> vorwä
rts
        moveq   #31,d3                  doppelt von 2.
reihe
        moveq   #61,d5                  5.reihe für ep
        bra.s   .b2
.b1     moveq   #-10,d2                 schwarz -> rüc
kwärts
        moveq   #81,d3                  doppelt von 7.
reihe
        moveq   #51,d5                  4.reihe für ep
.b2     tst.b   (a2,d2.w)
        bne.s   .bs
        move.b  d1,(a0)+                einfachen Baue
rnzug
        move    d1,d4
        add     d2,d4
        move.b  d4,(a0)+                eintragen
        cmp.b   d3,d1                   doppelt möglic
h?
        blt.s   .bs
        addq    #8,d3
        cmp.b   d3,d1
        bge.s   .bs
        move    d2,d3
        add     d2,d3                   Distanz für do
ppelt
        tst.b   (a2,d3.w)
        bne.s   .bs
        move.b  d1,(a0)+                doppelten Baue
rnzug
        move    d1,d4
        add     d3,d4
        move.b  d4,(a0)+                eintragen
.bs     subq    #1,d2                   schlagen ??
        moveq   #1,d3                   zwei Möglichke
iten
.bs1    move.b  (a2,d2.w),d4
        beq.s   .bsx
```

```
                bmi.s     .bsl
                tst.b     d0                    Farbe am Zug
                beq.s     .bsx
                bra.s     .bsw
        .bsl    tst.b     d0
                bne.s     .bsx
                cmp.b     #255,d4               Rand ?
                beq.s     .bsx

        .bsw    move.b    d1,(a0)+              schlagenden Ba
        uernzug
                move      d1,d4
                add       d2,d4
                move.b    d4,(a0)+              eintragen

        .bsx    addq      #2,d2                 andere Richtun
        g
                dbf       d3,.bsl
                subq      #3,d2

                cmp.b     d5,d1                 reihe für ep
                blt.s     .ep2
                addq      #7,d5
                cmp.b     d5,d1
                bgt.s     .ep2

                moveq     #0,d4
                move.b    d1,d4                 en passant ???
                lea       xmodtab(pc),a2
                move.b    -20(a2,d4.w),d4
                cmp.b     #8,d4
                bge.s     .epl
                btst      d4,(a4)
                beq.s     .epl

                move.b    d1,(a0)+
                move      d1,d3
                add       d2,d3
                addq      #1,d3
                move.b    d3,(a0)+

        .epl    subq.b    #2,d4                 ep links
                bmi.s     .ep2
                btst      d4,(a4)
                beq.s     .ep2

                move.b    d1,(a0)+
                move      d1,d3
                add       d2,d3
                subq      #1,d3
                move.b    d3,(a0)+

        .ep2    bra       .weiter

        .zkl    lea       rosl(a5),a4
                bra.s     .zkb

        .zk     tst.b     d0
                bne.s     .zkl
                lea       rowl(a5),a4

        .zkb    tst       (a4)
                beq.s     .zkx

                tst.b     (a4)
                beq.s     .zkwl
                tst       -3(a5,d1)
                bne.s     .zkwl
                tst.b     -1(a5,d1)
                bne.s     .zkwl

                move.b    d1,(a0)+              lange Rochade
                move.b    d1,d4
                sub.b     #2,d4
                move.b    d4,(a0)+

        .zkwl   tst.b     1(a4)
                beq.s     .zkx
                tst       1(a5,d1)
                bne.s     .zkx

                move.b    d1,(a0)+              kurze Rochade
                move.b    d1,d4
                add.b     #2,d4
                move.b    d4,(a0)+

        .zkx    lea       adkk(pc),a1           Richtungskonst
        anten König
                bra.s     .zks
        .zs     lea       adks(pc),a1           Richtungskonst
        anten Springer
        .zks    moveq     #7,d2
        .sil    move      (a1)+,d3
                move.b    (a2,d3.w),d4
                beq.s     .siw
                bmi.s     .ksl
                tst.b     d0                    Farbe am Zug
                beq.s     .six
                bra.s     .siw
        .ksl    tst.b     d0
                bne.s     .six
                cmp.b     #255,d4               Rand ?
                beq.s     .six
```

Listing SCHACH

```

        .siw    move.b    d1,(a0)+              einfachen Zug
                move      d1,d4
                add       d3,d4
                move.b    d4,(a0)+              eintragen

        .six    dbf       d2,.sil               nächste Richtu
        ng
                bra.s     .weiter

        .zl     lea       adkl(pc),a1           Richtungskonst
        anten Läufer
                moveq     #3,d2                 4 Richtungen
                bra.s     .zdtl
        .zd     lea       adkd(pc),a1           Richtungskonst
        anten Dame
                moveq     #7,d2                 8 Richtungen
                bra.s     .zdtl
        .zt     lea       adkt(pc),a1           Richtungskonst
        anten Turm
                moveq     #3,d2                 4 Richtungen
        .zdtl
        .mul    move      (a1)+,d5
                move      d5,d3
        .mu2    move.b    (a2,d3.w),d4
                beq.s     .muw
                bmi.s     .kml
                tst.b     d0                    Farbe am Zug
                beq.s     .mux
                bra.s     .muw
        .kml    tst.b     d0
                bne.s     .mux
                cmp.b     #255,d4
                beq.s     .mux

        .muw    move.b    d1,(a0)+              +Zug
                move      d1,d6
                add       d3,d6
                move.b    d6,(a0)+              eintragen

                tst.b     d4                    freies Feld ?
                bne.s     .mux
                add       d5,d3                 wenn ja, weite
        r testen
                bra.s     .mu2

        .mux    dbf       d2,.mul               nächste Richtu
        ng

        .weiter addq      #1,d1                 nächstes Feld
                dbf       d7,.loop

                move.l    (a7)+,a1
                sub.l     a1,a0
                move.l    a0,d1                 Länge der Zugl
        iste
                lsr       #1,d1                 Anzahl Züge

                move      d1,.zanz
                move      d1,-(a7)

                subq      #1,d1
        .sol    bsr       nbrett
                move.b    (a1)+,d3
                move.b    (a1)+,d2

                tst.b     d0
                bne.s     .ros
                cmp.b     120(a5),d3
                bne.s     .nro
                bra.s     .roc
        .ros    cmp.b     136(a5),d3
                bne.s     .nro
        .roc    move.b    d3,d4
                sub.b     d2,d4
                cmp.b     #2,d4
                beq.s     .rtst
                cmp.b     #-2,d4
                bne.s     .nro

        .rtst   movem     d2-d3,-(a7)
                bsr.s     chtest
                bne.s     .kill
                movem     (a7)+,d2-d3

        .nro    bsr       zug
                bsr.s     chtest
                beq.s     .so2
        .sox    clr       -2(a1)
                subq      #1,.zanz
                beq.s     .nix
        .so2    lea       -160(a5),a5
                dbf       d1,.sol

                move      (a7)+,d0
                rts                             Flags für Anza
        hl gesetzt

        .nix    lea       -160(a5),a5
                addq      #2,a7
                moveq     #0,d0                 alle Züge gest
        richen
                rts

```

```
.kill   addq.l  #4,a7
        bra.s   .sox
.jmptab dc.l    .zb,.zs,.zl,.zt,.zd,.zk
.zanz   ds.w    1
chtest  tst.b   d0
        bne.s   .chs
        move.b  120(a5),d4
        moveq   #-128,d5
        bra.s   .chx
.chs    move.b  136(a5),d4
        moveq   #0,d5
.chx    ext.w   d4
        lea     (a5,d4.w),a2
        moveq   #2,d6
        or      d5,d6
        lea     adks(pc),a0
        moveq   #7,d7
.csl    move    (a0)+,d3
        move.b  (a2,d3.w),d3
        cmp.b   d6,d3
        beq.s   .schach
        dbf     d7,.csl
        moveq   #6,d6
        or      d5,d6
        lea     adkk(pc),a0
        moveq   #7,d7
.ckl    move    (a0)+,d3
        move.b  (a2,d3.w),d3
        cmp.b   d6,d3
        beq.s   .schach
        dbf     d7,.ckl
        tst.b   d0
        bne.s   .cbs
        cmp.b   #129,9(a2)
        beq.s   .schach
        cmp.b   #129,11(a2)
        beq.s   .schach
        bra.s   .cnb
.cbs    cmp.b   #1,-9(a2)
        beq.s   .schach
        cmp.b   #1,-11(a2)
        beq.s   .schach
.cnb    lea     adkt(pc),a0
        moveq   #4,d6
        bsr.s   .cm
        lea     adkl(pc),a0
        moveq   #3,d6
        bsr.s   .cm
        moveq   #0,d7
        rts
.schach moveq   #-1,d7
        rts
.cm     move    d0,-(a7)
        or      d5,d6
        moveq   #5,d0
        or      d5,d0
        moveq   #3,d7
.mul    move    (a0)+,d2
        move    d2,d3
.mu2    move.b  (a2,d3.w),d4
        beq.s   .muw
        cmp.b   d6,d4
        beq.s   .sch2
        cmp.b   d0,d4
        beq.s   .sch2
.mun    dbf     d7,.mul
        move    (a7)+,d0
        rts
.muw    add     d2,d3
        bra.s   .mu2
.sch2   move    (a7)+,d0
        addq.l  #4,a7
        bra.s   .schach
nbrett  move.b  120(a5),160+120(a5)      weißer König
        move.b  136(a5),160+136(a5)      schwarzer
        move.l  152(a5),160+152(a5)      Rochade
        move.l  156(a5),160+156(a5)      Material, en p
assant
        lea     20(a5),a5
        lea     160(a5),a0
        rept    20
        move.l  (a5)+,(a0)+
        endr
        lea     60(a5),a5
        rts
```

Listing SCHACH

```
kill
show    movem   d0-d1,-(a7)              Feld normal da
rstellen
        add     d1,d0
        and.l   #1,d0
        bra.s   disp
mark    movem   d0-d1,-(a7)              Feld markieren
        moveq   #2,d0
disp    move.l  GfxBase(pc),a6
        move.l  rp(pc),a1
        jsr     SetAPen(a6)
        movem   (a7),d0-d1
        lsl     #5,d0
        add     #16,d0
        ext.l   d0
        move.l  d0,d2
        add.l   #31,d2
        mulu    #26,d1
        neg.l   d1
        add.l   #198,d1
        move.l  d1,d3
        add.l   #25,d3
        move.l  rp(pc),a1
        jsr     RectFill(a6)
        moveq   #0,d1
        movem   (a7)+,d2-d3
        move    d3,d4
        mulu    #10,d4
        add     d2,d4
        move.b  21(a5,d4.w),d0
        beq.s   .12
        bpl.s   .11
        moveq   #1,d1
.11     and     #$007f,d0
        subq    #1,d0
        bsr.s   copy
.12     rts
copy    lsl     #2,d0
        mulu    #1280,d1
        add     d1,d0
        move    #7280+642,d1
        lsl     #2,d2
        add     d2,d1
        mulu    #1040,d3
        sub     d3,d1
        moveq   #25,d7
.11     lea     fig0(pc),a0
        lea     plane0(pc),a1
        move.l  (a0)+,a2
        move.l  (a2,d0.w),d2
        moveq   #3,d6
.12     move.l  (a0)+,a2
        move.l  (a2,d0.w),d5
        or.l    d5,d2
        dbf     d6,.12
        not.l   d2
        lea     fig0(pc),a0
        moveq   #4,d6
.13     move.l  (a1)+,a3
        and.l   d2,(a3,d1.w)
        move.l  (a0)+,a2
        move.l  (a2,d0.w),d5
        or.l    d5,(a3,d1.w)
        dbf     d6,.13
        add     #40,d0
        add     #40,d1
        dbf     d7,.11
        rts
getm    move.l  a5,-(a7)
.gl     bsr.s   readmen
        bmi.s   .gm
        bsr     menue
.gm     btst.b  #6,$bfe001
        bne.s   .gl
        move.l  mx(pc),a0
        move.l  my(pc),a1
        moveq   #0,d1
        move    (a0),d0
        move    (a1),d1
        sub     #16,d0
        lsr     #5,d0
        cmp     #7,d0
        bgt     .gl
        sub     #16,d1
        divu    #26,d1
        cmp     #7,d1
        bgt     .gl
        not     d1
        and     #7,d1
```

```
            move.l  (a7)+,a5
            rts
readmen     move.l  4.w,a6
            move.l  window(pc),a0
            move.l  $56(a0),a0          Userport
            jsr     GetMsg(a6)
            tst.l   d0
            beq.s   readmen
            move.l  d0,a1
            move.l  $14(a1),d4          msgtype
            move    $18(a1),d5          code
            jsr     ReplyMsg(a6)
            cmp.l   #$100,d4            menupick
            bne.s   .no
            cmp     #$ffff,d5
            beq.s   .no
            move    d5,d6
            and     #$1f,d5             menu-nr
            lsr     #5,d6
            move    d6,d7
            and     #$3f,d6             item-nr
            lsr     #6,d7               sub-item
            rts
.no         moveq   #-1,d5
            rts
menue       tst     d5
            bne.s   .men1
            tst     d6
            beq.s   .exit
            bra     abbruch             Ende
.men1       cmp     #1,d5               Datei
            bne.s   .men2
            tst     d6
            beq.s   load
            bra.s   save
.men2       tst     d6
            bne.s   .it2
            bsr     setlev
            bra.s   .exit
.it2        cmp     #1,d6
            bne     neustrt             Neustart
            bra     change              Seitenwechsel
.exit       rts
load        move.l  dosbase(pc),a4
            lea     Read(a4),a3
            move.l  #1005,d2
            bsr.s   ls
            tst.l   d7
            beq.s   .exit
            bsr     setup               Brett darstell
en
            move    ltiefe(pc),d7
            subq    #1,d7
            bsr     setlev
            bra     weiter
.exit       rts
save        move.l  dosbase(pc),a4
            lea     Write(a4),a3
            move.l  #1006,d2
ls          bsr     menuoff
            move    #1,vbllock
            bsr     getname
            tst.b   dobuff
            beq.s   .exit
            move.l  feldbuf(pc),a5
            move.l  #dobuff,d1
            jsr     Open(a4)
            move.l  d0,d7
            beq.s   .error
            move.l  d7,d1
            move.l  a5,d2               Brett
            move.l  #160,d3
            jsr     (a3)
            move.l  d7,d1
            move.l  #saveanf,d2
            moveq   #saveend-saveanf,d3
            jsr     (a3)
            move.l  d7,d1
            jsr     Close(a4)
            clr     vbllock
            bsr     menuon
            rts
```

Listing SCHACH

```
.error      lea     .ertxt(pc),a0
            move    #146,d0
            move    #240,d1
            moveq   #16,d2
            bsr     prtxt
.exit       clr     vbllock
            bsr     menuon
            rts
.ertxt      dc.b    "Datei-Fehler !!!"
getname     movem.l d0-d7/a0-a6,-(a7)
            move.l  IntBase,a6
            lea     windat2,a0
            jsr     OpenWindow(a6)
            move.l  d0,-(a7)
            move.l  d0,a0
            lea     gadget0,a1
            moveq   #-1,d0
            jsr     AddGadget(a6)
            lea     gadget0,a0
            move.l  (a7),a1
            sub.l   a2,a2
            jsr     RefreshGadgets(a6)
            lea     gadget0,a0
            move.l  (a7),a1
            sub.l   a2,a2
            jsr     ActivateGadget(a6)
            move.l  4.w,a6
.gn1        move.l  (a7),a0
            move.l  $56(a0),a0          Userport
            jsr     GetMsg(a6)
            tst.l   d0
            beq.s   .gn1
            move.l  d0,a1
            move.l  $14(a1),d4          msgtype
            move    $18(a1),d5          code
            jsr     ReplyMsg(a6)
            cmp.l   #$40,d4
            bne.s   .gn1
            move.l  (a7),a0
            move.l  IntBase(pc),a6
            lea     gadget0(pc),a1
            jsr     RemoveGadget(a6)
            move.l  (a7)+,a0
            jsr     CloseWindow(a6)
            movem.l (a7)+,d0-d7/a0-a6
            rts
setup       moveq   #7,d1
.s1         moveq   #7,d0
.s2         movem   d0-d1,-(a7)
            bsr     show
            movem   (a7)+,d0-d1
            dbf     d0,.s2
            dbf     d1,.s1
            rts
waitm       move.l  d0,-(a7)
            move.l  #10000,d0
.wm1        subq.l  #1,d0
            bne.s   .wm1
.wm         btst.b  #6,$bfe001          warten bis Mau
staste losgelassen
            beq.s   .wm
            move.l  #10000,d0
.wm2        subq.l  #1,d0
            bne.s   .wm2
            move.l  (a7)+,d0
            rts
brett       move.l  GfxBase(pc),a6
            move.l  rp(pc),a1
            moveq   #2,d0
            jsr     SetAPen(a6)
            move.l  rp(pc),a1
            moveq   #15,d0
            move.l  d0,d1
            jsr     Move(a6)
            move.l  rp(pc),a1
            moveq   #15,d0
            lea     ptab(pc),a0
            jsr     PolyDraw(a6)
            move.l  rp(pc),a1
            moveq   #0,d0
```

```
        jsr     SetBPen(a6)

        move    #"A8",st1

        moveq   #7,d7
        moveq   #28,d6
        moveq   #30,d5
.14     move.l  rp(pc),a1
        move.l  d6,d0
        moveq   #9,d1
        jsr     Move(a6)

        move.l  rp(pc),a1
        lea     st1(pc),a0
        moveq   #1,d0
        jsr     Text(a6)

        move.l  rp(pc),a1
        move.l  d5,d1
        moveq   #2,d0
        jsr     Move(a6)

        move.l  rp(pc),a1
        lea     st2(pc),a0
        moveq   #1,d0
        jsr     Text(a6)

        add     #$ff,st1

        add.l   #32,d6
        add.l   #26,d5

        dbf     d7,.14

        rts

st1     ds.b    1
st2     ds.b    1

ptab    dc.w    15,15,272,15,272,224,15,224,15,15
        dc.w    14,14,273,14,273,225,14,225,14,14
        dc.w    13,13,274,13,274,226,13,226,13,13

iniscrn move.l  IntBase(pc),a6
        lea     xpos(pc),a0             Screen-Tabelle
        jsr     OpenScreen(a6)
        move.l  d0,screen
        move.l  d0,screen2
        move.l  d0,a1
        move.l  $c0(a1),plane0          Plane 0
        move.l  $c4(a1),plane0+4
        move.l  $c8(a1),plane0+8
        move.l  $cc(a1),plane0+12
        move.l  $d0(a1),plane0+16
        lea     $2c(a1),a0
        move.l  a0,vp
        lea     $10(a1),a0
        move.l  a0,my
        lea     $12(a1),a0
        move.l  a0,mx

        lea     windat(pc),a0
        jsr     OpenWindow(a6)
        move.l  d0,window
        move.l  d0,a1
        move.l  50(a1),rp               RastPort

        move.l  d0,a0
        lea     menu1(pc),a1
        jsr     SetMenuStrip(a6)

        rts

clrscrn move.l  IntBase(pc),a6
        move.l  window(pc),a0
        jsr     CloseWindow(a6)
        move.l  screen(pc),a0
        jsr     CloseScreen(a6)
        rts

init    move.l  4.w,a6
        lea     intname(pc),a1
        jsr     OpenLib(a6)
        move.l  d0,IntBase

        move.l  #64000,d0
        moveq   #0,d1
        jsr     AllocMem(a6)
        move.l  d0,puffer

        move.l  #12800,d0
        moveq   #0,d1
        jsr     AllocMem(a6)
        moveq   #4,d7
        lea     fig0(pc),a0
.11     move.l  d0,(a0)+
        add.l   #2560,d0
        dbf     d7,.11

        move.l  #160*10,d0
        moveq   #0,d1
        jsr     AllocMem(a6)
        move.l  d0,feldbuf

        move.l  d0,a0
        move    #379,d0
        moveq   #-1,d1
.12     move.l  d1,(a0)+
        dbf     d0,.12

        move.l  #1024*10,d0             10 Halbzüge
        moveq   #0,d1
        jsr     AllocMem(a6)
        move.l  d0,zliste

        move.l  #8*11,d0                10 Halbzüge
        moveq   #0,d1
        jsr     AllocMem(a6)
        move.l  d0,font

        lea     grapnam(pc),a1
        jsr     OpenLib(a6)
        move.l  d0,GfxBase

        lea     dosname(pc),a1
        jsr     OpenLib(a6)
        move.l  d0,dosbase

        rts

clear   move.l  4.w,a6
        move.l  GfxBase(pc),a1
        jsr     CloseLib(a6)
        move.l  IntBase(pc),a1
        jsr     CloseLib(a6)
        move.l  dosbase(pc),a1
        jsr     CloseLib(a6)

        move.l  #64000,d0
        move.l  puffer(pc),a1
        jsr     FreeMem(a6)
        move.l  #12800,d0
        move.l  fig0(pc),a1
        jsr     FreeMem(a6)
        move.l  #160*10,d0
        move.l  feldbuf(pc),a1
        jsr     FreeMem(a6)
        move.l  #1024*10,d0
        move.l  zliste(pc),a1
        jsr     FreeMem(a6)
        move.l  #8*11,d0
        move.l  font(pc),a1
        jsr     FreeMem(a6)
        rts

loadpic move.l  dosbase(pc),a4

        move.l  #1005,d2
        move.l  #fignam,d1              Dateiname !
        jsr     Open(a4)
        move.l  d0,handle
        beq     error

        move.l  #64000,d3               Daten lesen
        move.l  handle(pc),d1
        move.l  puffer(pc),d2
        jsr     Read(a4)

        move.l  handle(pc),d1
        jsr     Close(a4)               Datei schliess
en

                                        * Farben ausle
sen
        move.l  puffer(pc),a1
        lea     farbtab(pc),a2
.11     addq.l  #2,a1
        cmp.l   #"CMAP",(a1)            Farb-Chunk suc
hen
        bne.s   .11
        addq.l  #8,a1
        moveq   #31,d7                  32 Farbregiste
r
.12     moveq   #0,d0
        moveq   #0,d1
        move.b  (a1)+,d0
        lsl     #4,d0
        or.b    (a1)+,d0                Farbwert zusam
mensetzen
        move.b  (a1)+,d1
        lsr     #4,d1
        or.b    d1,d0
        move    d0,(a2)+
        dbf     d7,.12

        bsr.s   getbit

        move.l  GfxBase(pc),a6
        move.l  vp(pc),a0
        lea     farbtab(pc),a1
        moveq   #32,d0
        jsr     LoadRGB4(a6)

        rts

```

Listing SCHACH

```
                                        * Nun die Bitmap
getbit  move.l  puffer(pc),a1
.l1     addq.l  #2,a1
        cmp.l   #"BODY",(a1)            Datenbeginn suchen
        bne.s   .l1
        addq.l  #8,a1                   Anzahl Datenbytes (gepackt)

        moveq   #57,d7                  Zeilen
        moveq   #0,d0                   Adresse der Zeile
.line   moveq   #4,d6                   Bitplanes
        lea     fig0(pc),a0             1.Plane
.plane  move.l  (a0)+,a2
        add.l   d0,a2                   Zeile in Plane
        clr     d1                      geschriebene Bytes in der Zeile
.iloop  clr     d2
        move.b  (a1)+,d2                Code-Byte
        tst.b   d2
        bmi.s   .packed
        add     d2,d1                   Bytes in dieser Zeile
        addq    #1,d1                   +1 fuer die Null
.unp    move.b  (a1)+,(a2)+             Byte uebernehmen
        dbf     d2,.unp
        bra.s   .cnt
.packed move.b  (a1)+,d3                Wiederhol-Byte
        neg.b   d2                      Vorzeichen weg
        add     d2,d1                   Bytes in dieser Zeile
        addq    #1,d1                   +1 fuer die Null
.pac    move.b  d3,(a2)+
        dbf     d2,.pac
.cnt    cmp     #40,d1                  Schon Zeilenende?
        blt.s   .iloop
        dbf     d6,.plane
        add.l   #40,d0                  naechste Zeile
        dbf     d7,.line
        rts

intname dc.b    "intuition.library",0
IntBase ds.l    1

grapnam dc.b    "graphics.library",0
GfxBase ds.l    1

dosname dc.b    "dos.library",0
dosbase ds.l    1

fignam  dc.b    "fig",0

handle  ds.l    1

xpos    dc.w    0                       Anfang Screen-Daten
ypos    dc.w    0
width   dc.w    320
heigth  dc.w    256
depth   dc.w    5
dp      dc.b    0
bp      dc.b    1
sview   dc.w    2
stype   dc.w    1
fontl   dc.l    0
title   dc.l    0
gadget  dc.l    0
bitmap  dc.l    0

windat  dc.w    0                       Window-Daten
        dc.w    0
        dc.w    320
        dc.w    256
        dc.b    1
        dc.b    0
        dc.l    $040108                 Window-Active, Menu-Pick, Maus
        dc.l    $1800                   Active(1) & Borderless(8)
        dc.l    0
        dc.l    0
        dc.l    0
screen  ds.l    1
        dc.l    0
        dc.w    10
        dc.w    10
        dc.w    320
        dc.w    256
        dc.w    15

windat2 dc.w    33                      Window-Daten
        dc.w    88
        dc.w    222
        dc.w    80
        dc.b    -1
        dc.b    -1
        dc.l    $040040                 Window-Active
        dc.l    $1000                   Active(1) & Borderless(8)
        dc.l    0                       erstes GadGet
        dc.l    0
        dc.l    0
screen2 ds.l    1
        dc.l    0
        dc.w    10
        dc.w    10
        dc.w    100
        dc.w    60
        dc.w    15

window  ds.l    1
rp      ds.l    1
vp      ds.l    1
plane0  ds.l    6
puffer  ds.l    1

menu1   dc.l    menu2
        dc.w    10,0,72,10
        dc.w    1                       menuenabled
        dc.l    titl1
        dc.l    item1
        dc.w    0,0,0,0

titl1   dc.b    " Projekt ",0

item1   dc.l    item2
        dc.w    -5,0,100,11
        dc.w    $92                     text=2 enable=10 highbox=80
        dc.l    0
        dc.l    itxt1
        dc.l    0
        dc.b    0,0
        dc.l    subitl
        dc.w    0

itxt1   dc.b    1,0,0,0
        dc.w    8,2
        dc.l    0
        dc.l    istr1
        dc.l    0

istr1   dc.b    "Info",0

subitl  dc.l    0
        dc.w    60,10,208,33
        dc.w    $92                     text=2 enable=10 highbox=80
        dc.l    0
        dc.l    stxt1
        dc.l    0
        dc.b    0,0
        dc.l    0
        dc.w    0

stxt1   dc.b    1,0,0,0
        dc.w    8,2
        dc.l    0
        dc.l    sstr1
        dc.l    stxt2
stxt2   dc.b    1,0,0,0
        dc.w    8,13
        dc.l    0
        dc.l    sstr2
        dc.l    stxt3
stxt3   dc.b    1,0,0,0
        dc.w    8,24
        dc.l    0
        dc.l    sstr3
        dc.l    0

sstr1   dc.b    "       Schach !!!",0
sstr2   dc.b    "von N.Nebel und M.Cordes",0
sstr3   dc.b    "Oktober 89, in Clausthal",0

item2   dc.l    0
        dc.w    -5,12,100,11
        dc.w    $92                     text=2 enable=10 highbox=80
        dc.l    0
        dc.l    itxt2
        dc.l    0
        dc.b    0,0
        dc.l    0
        dc.w    0

itxt2   dc.b    1,0,0,0
        dc.w    8,2
        dc.l    0
        dc.l    istr2
        dc.l    0

istr2   dc.b    "Ende",0

**********************************************************

menu2   dc.l    menu3
        dc.w    85,0,56,10
```

Listing SCHACH

```
            dc.w    1                       menuenabled
            dc.l    titl2
            dc.l    item3
            dc.w    0,0,0,0

titl2       dc.b    " Datei ",0

item3       dc.l    item4
            dc.w    -5,0,135,11
            dc.w    $92                     text=2 enable=10 highbox=80
            dc.l    0
            dc.l    itxt3
            dc.l    0
            dc.b    0,0
            dc.l    0
            dc.w    0

itxt3       dc.b    1,0,0,0
            dc.w    8,2
            dc.l    0
            dc.l    istr3
            dc.l    0

istr3       dc.b    "Spiel laden",0

item4       dc.l    0
            dc.w    -5,12,135,11
            dc.w    $92                     text=2 enable=10 highbox=80
            dc.l    0
            dc.l    itxt4
            dc.l    0
            dc.b    0,0
            dc.l    0
            dc.w    0

itxt4       dc.b    1,0,0,0
            dc.w    8,2
            dc.l    0
            dc.l    istr4
            dc.l    0

istr4       dc.b    "Spiel speichern",0

**********************************************************

menu3       dc.l    0
            dc.w    145,0,56,10
            dc.w    1                       menuenabled
            dc.l    titl3
            dc.l    item5
            dc.w    0,0,0,0

titl3       dc.b    " Spiel ",0

item5       dc.l    item6
            dc.w    -5,0,115,11
            dc.w    $92                     text=2 enable=10 highbox=80
            dc.l    0
            dc.l    itxt5
            dc.l    0
            dc.b    0,0
            dc.l    subit2
            dc.w    0

itxt5       dc.b    1,0,0,0
            dc.w    8,2
            dc.l    0
            dc.l    istr5
            dc.l    0

istr5       dc.b    "Spielstufe",0

subit2      dc.l    subit3
            dc.w    110,10,22,11
            dc.w    $92                     text=2 enable=10 highbox=80
            dc.l    0
            dc.l    stxt4
            dc.l    0
            dc.b    0,0
            dc.l    0
            dc.w    0

subit3      dc.l    subit4
            dc.w    110,21,22,11
            dc.w    $92                     text=2 enable=10 highbox=80
            dc.l    0
            dc.l    stxt5
            dc.l    0
            dc.b    0,0
            dc.l    0
            dc.w    0

subit4      dc.l    0
            dc.w    110,32,22,11
            dc.w    $92                     text=2 enable=10 highbox=80
```

Listing SCHACH

```
            dc.l    0
            dc.l    stxt6
            dc.l    0
            dc.b    0,0
            dc.l    0
            dc.w    0

stxt4       dc.b    1,0,0,0
            dc.w    8,2
            dc.l    0
            dc.l    sstr4
            dc.l    0
stxt5       dc.b    1,0,0,0
            dc.w    8,2
            dc.l    0
            dc.l    sstr5
            dc.l    0
stxt6       dc.b    1,0,0,0
            dc.w    8,2
            dc.l    0
            dc.l    sstr6
            dc.l    0

sstr4       dc.b    "1 ",0
sstr5       dc.b    "2 ",0
sstr6       dc.b    "3 ",0

item6       dc.l    item7
            dc.w    -5,12,115,11
            dc.w    $92                     text=2 enable=10 highbox=80
            dc.l    0
            dc.l    itxt6
            dc.l    0
            dc.b    0,0
            dc.l    0
            dc.w    0

itxt6       dc.b    1,0,0,0
            dc.w    8,2
            dc.l    0
            dc.l    istr6
            dc.l    0

istr6       dc.b    "Seitenwechsel",0

item7       dc.l    0
            dc.w    -5,24,115,11
            dc.w    $92                     text=2 enable=10 highbox=80
            dc.l    0
            dc.l    itxt7
            dc.l    0
            dc.b    0,0
            dc.l    0
            dc.w    0

itxt7       dc.b    1,0,0,0
            dc.w    8,2
            dc.l    0
            dc.l    istr7
            dc.l    0

istr7       dc.b    "Neustart",0

gadget0     dc.l    0                       GadGet-Daten
            dc.w    12                      links oben
            dc.w    45
            dc.w    180                     Breite
            dc.w    12                      Höhe
            dc.w    0
            dc.w    $0201                   Stringcenter
            dc.w    $04                     StringGadGet
            dc.l    border0                 Border
            dc.l    0
            dc.l    name0                   Textstruktur
            dc.l    0                       Mutual Exclude
            dc.l    info0                   Special Info
            dc.w    0                       ID
            dc.l    0

cord0       dc.w    0,0
            dc.w    202,0
            dc.w    202,11
            dc.w    0,11
            dc.w    0,0

border0     dc.w    -2                      links oben
            dc.w    -2
            dc.b    1                       Pen
            dc.b    0
            dc.b    0                       DrawMode
            dc.b    5                       5 Paare
            dc.l    cord0
            dc.l    0

info0       dc.l    dobuff                  StringInfo
            dc.l    undbuf
            dc.w    0                       Buffer Position
            dc.w    80                      Größe
```

Listing SCHACH

```
         dc.w    0                     Position Cursor-Zeichen
         ds.l    10
name0    dc.b    1                     IntuiText
         dc.b    0                     Hintergrund
         dc.b    0                     Drawmode
         ds.b    1
         dc.w    0                     Lage links oben
         dc.w    -20
         dc.l    0                     Fontstruktur
         dc.l    gadtxt
         dc.l    0
gadtxt   dc.b    "Dateinamen eingeben :",0
dobuff   dc.b    "Partie01",0
         ds.b    71
undbuf   ds.b    80
farbtab  ds.w    32
fig0     ds.l    5
feldbuf  ds.l    1
my       ds.l    1
mx       ds.l    1
adkt
adkk
adkd     dc.w    1,10,-1,-10           gerade
adkl     dc.w    9,11,-9,-11           diagonal
adks     dc.w    8,12,19,21
         dc.w    -8,-12,-19,-21
werte    dc.w    $100,$300,$300,$500,$900,0
xmodtab  dc.b    0,1,2,3,4,5,6,7,8,9   x mod 10
         dc.b    0,1,2,3,4,5,6,7,8,9
         dc.b    0,1,2,3,4,5,6,7,8,9
         dc.b    0,1,2,3,4,5,6,7,8,9
         dc.b    0,1,2,3,4,5,6,7,8,9
         dc.b    0,1,2,3,4,5,6,7,8,9
         dc.b    0,1,2,3,4,5,6,7,8,9
         dc.b    0,1,2,3,4,5,6,7,8,9
xbittab  dc.b    $00,$01,$02,$04,$08,$10,$20,$40,$80,$00   ep-bits
         dc.b    $00,$01,$02,$04,$08,$10,$20,$40,$80,$00
         dc.b    $00,$01,$02,$04,$08,$10,$20,$40,$80,$00
         dc.b    $00,$01,$02,$04,$08,$10,$20,$40,$80,$00
         dc.b    $00,$01,$02,$04,$08,$10,$20,$40,$80,$00
         dc.b    $00,$01,$02,$04,$08,$10,$20,$40,$80,$00
         dc.b    $00,$01,$02,$04,$08,$10,$20,$40,$80,$00
         dc.b    $00,$01,$02,$04,$08,$10,$20,$40,$80,$00
tiefe    ds.w    1
saveanf
ltiefe   ds.w    1
spieler  ds.b    1
saveend
bspiel   ds.b    1
zliste   ds.l    1
epflag   ds.b    1
roflag   ds.b    1
vbllock  ds.w    1
stackp   ds.l    1
debug    dc.l    0
count    ds.w    1
min      ds.w    1
sek      ds.w    1
font     ds.l    1
intstr   dc.l    0,0
         dc.b    2                     Typ
         dc.b    -100                  Prio
         dc.l    0
         dc.l    0
         dc.l    vbl
```

Listing SCHACH

Gewußt wie!

Wissenswertes in Kurzform

Unser Treffpunkt für Kurztips rund um den Amiga entwickelt sich langsam, aber stetig. Hier nun finden Sie ein volles Paket an Informationen und Wissenswertem zu BASIC, CLI und allem, was den Amiga ausmacht. Fangen wir am besten gleich mal an.

Zeichenketten sortieren

Das folgende kleine Programm sortiert Zeichenketten nach ASCII-Codes (alphabetisch) in BASIC:

```
FOR i=n-1 TO 1 STEP -1
  FOR j=1 TO i
    IF so$(j)>so$(j+1) THEN
      SWAP so$(j),so$(j+1)
  NEXT j
NEXT i
```

Die Variable 'n' steht für die Anzahl der Zeichenketten, die in so$(?) abgelegt sind. Nach dem Sortieren enthält so$(n) die höchste Zeichenkette. Durch Vertauschen des Zeichens > mit < erhält man außerdem das umgekehrte Ergebnis.

(Thomas Renner/jb)

Preferences-Trick

Besitzt man nur ein Laufwerk und möchte auf einer Boot-Diskette, auf der das 'Preferences'-Tool fehlt, dennoch die Grundeinstellungen (Drucker, Farben, Bildschirmposition) ändern, so verfährt man wie folgt:

1.) Nach einem RESET oder nach Anschalten des Computers verlangt dieser die Workbench. Jetzt sollte die Boot-Diskette eingelegt werden.

2.) Während des Bootens CTRL-D drücken, um die Startup-Sequence zu unterbrechen, danach die Workbench-Diskette einlegen.

3.) Im CLI (woran man sich jetzt befindet) eingeben: *DF0:Preferences*, dadurch wird das Tool gestartet, was sich auf der jetzt eingelegten Workbench befindet. Ist das Tool gestartet worden, verlangt der Amiga die erste Diskette wieder. Jetzt können die Voreinstellungen geändert werden, die dann beim nächsten Booten vorliegen.

(Robert Münch/jb)

Sparsam ist nützlich...

Jeder, der schon des öfteren CLI-Befehle oder Dateien in die RAM-Disk kopiert hat, weiß, wie mühsam es ist, für jeden Befehl, der übergeben werden soll, eine eigene Befehlszeile zu schreiben:

COPY C:#?.info RAM:
COPY C:Schublaedchen RAM:
usw.

Es gibt nun zwei Möglichkeiten, dieses Problem zu lösen:

1.) Wenn man genau weiß, daß man immer wieder dieselben Befehle in den RAM-Speicher laden will, ist es zweckmäßig, die Befehlszeilen mit einem Editor (ED leistet hier gute Dienste) zusammenzufassen und dann die Datei unter einem Namen (z.B.'Befehle') abzuspeichern. Jetzt braucht man nur noch im CLI 'EXECUTE (Dateiname)' einzugeben, und die Ausführung findet statt. Werden die Befehlsfolgen in der Startup-Sequence vor dem Befehl 'ENDCLI >NIL:' eingefügt, findet das Kopieren nach dem Start direkt statt.

2.) Wenn Sie nicht genau wissen, welche Befehle Sie jeweils im RAM brauchen, können Sie ein Befehlsmakro dafür erstellen, und zwar folgendermaßen:

a) Öffnen der Shell-Startup mit 'ed s/shell-startup'.
b) Die Zeile 'alias coRam copy [] Ram: ' anfügen.
c) Speichern der Shell-Startup
d) Den Shell-Bildschirm verlassen und anschließend wieder neu starten.

Punkt d) ist notwendig, um die Makrodefinition wirksam zu machen, Punkt b) ist der wichtigste, denn hier wird das Befehlsmakro erstellt, das heißt, hier wird eine Befehlszeile zu einem einfachen Befehlsausdruck zusammengefaßt. Statt dem langwierigen 'Copy ... TO ...' geben Sie jetzt einfach ein 'CoRam'. Die Klammern stehen für eine Eingabe, die der Computer an dieser Stelle erwartet.

Welche der Möglichkeiten Sie anwenden, wissen Sie sicher am besten, ich hoffe jedenfalls, Ihnen ein wenig lästige Tipparbeit abgenommen zu haben, denn 'time is cash'.

Diese Hilfe ist allerdings nur unter der Workbench 1.3 möglich.

'DiscDoctor' nebenbei

"Error validating Disk DF0:Disk is unreadable" – kommt als erstes. "Disc structure corrupt – Use DISCDOCTOR to correct ist" ist dann das (fast) endgültige Aus für defekte Disketten. Die Möglichkeit, den DiscDoctor mit 'RUN DiscDoctor' im Hintergrund laufen zu lassen, funktioniert leider nicht. Öffnen Sie statt dessen ein weiteres CLI-Fenster und starten Sie dort das Programm. Dort wird die defekte Diskette dann bearbeitet, während Sie Ihrer Beschäftigung nachgehen können.

(Andreas Bender/jb)

Nachtrag zum 'CLI-Fensterln' (AMIGA DOS 1/90)

Der Tip, einen NEWCLI-Befehl in seine Startup-Sequence einzubauen, ist schon o.k. Ich halte es auf meiner Maschine fast genauso, jedoch

ESCAPE-Sequencen für den CLI!

Die in < > gesetzten Zeichen sind die Platzhalter für die Werte.

ESC[<n>u	n Zeichen pro Zeile
ESC[<n>x	linker Rand in Pixeln
ESC[<n>y	oberer Rand in Pixeln
ESC[<n>t	Anzahl der Zeilen
ESC[c	Reset (wieder normale Werte)
ESC[0<SPACE>p	Cursor aus
ESC[<SPACE>p	Cursor ein
ESC[<y>;<x>H	Cursor an Position (x,y) setzen
ESC[<n>A	Cursor n Zeilen hoch
ESC[<n>B	Cursor n Zeilen runter
ESC[<n>C	Cursor n Zeichen rechts
ESC[<n>D	Cursor n Zeichen links
ESC[J	Fenster ab Cursorposition löschen
ESC[K	Zeile ab Cursorposition löschen

ESC[<a>m	Setzen von Schriftstil und Farbe
ESC[<a>;<b>m	
ESC[<a>;<b>;<c>m	(Werte siehe unten)
a	= 0 normal
	1 fett
	2 schwarz
	3 kursiv
	4 unterstrichen
	7 invers
b	= 30,31,32,33 (Vordergrund blau, weiß, schwarz, orange bei Grundfarben)
c	= 40,41,42,43 (Hintergrund blau, weiß, schwarz, orange bei Grundfarben)

CTRL-Sequencen für den CLI!

CTRL H	Zeichen löschen
CTRL I	Tabulator
CTRL M	Return
CTRL N	alternativer Zeichensatz
CTRL O	originaler Zeichensatz
CTRL L	Bildschirm löschen
CTRL G	Bildschirmblitz
CTRL J	Cursor runter
CTRL K	Cursor hoch
Left-Amiga V	Retry im Requester
Left-Amiga B	Cancel im Requester

Bei Verwendung des "ECHO"-Befehls muß ESC durch *e ersetzt werden!

(Norbert Dimpfl)

Abb.1: Die CTRL- und ESC-Sequenzen unter dem CLI hat Norbert Dimpfl für uns zusammengetragen

mit zwei wichtigen Unterschieden. Zum einen benutze ich unter der Version 1.3 natürlich die 'Shell' statt des einfachen CLI, zum anderen finde ich andere Fensterabmessungen praktischer: Oben will ich unter der Screen-Titelleiste gerade soviel freien Workbenchplatz lassen, daß ich dort hineinklicken kann, um bequem den freien Speicherplatz angezeigt zu bekommen. Und rechts möchte ich soviel Platz freilassen, daß die Diskettenpiktogramme sichtbar bleiben, um auch sie bequem und ohne große Fensterschiebereien anklicken zu können. Zusammen ergibt das folgenden Aufruf:

newshell
newcon:0/15/580/240/
AmigaShell from
s:Shell-Startup

Ein weiterer Tip betrifft die Multitaskingfähigkeit des Amiga. Wenn Sie zwei Laufwerke besitzen, können Sie einem ignoranten PC-Besitzer mal sehr handfest vorführen, was Multitasking ist: Formatieren Sie einfach einmal zwei Disketten gleichzeitig! Dazu öffnen Sie am besten zwei CLI- bzw. Shell-Fenster, und in beiden geben Sie den Befehl zum Formatieren einer Diskette in dem jeweiligen Laufwerk an. Ein PC dürfte damit wegen der dort üblichen Floppy-Controller-ICs sogar unter dem Betriebssystem OS/2 überfordert sein. Daneben ist dies auch ein ganz praxisgerechter Tip, wie man schnell und elegant viele Disketten formatieren kann.

(Dr. Peter Kittel/jb)

Noch mehr zum CLI

Automatisches Ausführen eines Befehls beim Öffnen des CLI-Windows.
Der Command-Line-Interpreter (CLI) des Amiga kann nach dem Anklicken des CLI-Icons einen Befehl automatisch ausführen. Dazu müssen wir das File "CLI" auf der Workbenchdiskette verändern.

Wir laden das Programm mit Hilfe eines Filemonitors (z.B. 'NewZap' aus der Public Domain) und suchen nach dem String "NewCLI". Dort müßte dann folgende Zeile stehen:

SYS: Workbench
CON:0/50/640/80/New
Cli Window

Interessant ist für uns der String "New CLI Window". Das ist der Titel des CLI-Fensters. Wenn wir hier nun einen String eingeben, der länger als 16 Zeichen ist, aber kleiner als 20 Zeichen sein muß, so werden die verbleibenden Zeichen als Befehl ausgeführt. Das eröffnet uns nun zwei verschiedene Möglichkeiten, die aber beide zum Erfolg führen:

Nehmen wir an, wir möchten nach Anklicken des CLI-Icons sofort den INFO-Befehl ausgeführt haben. Wir können den Infobefehl mit RENAME umbenennen, wobei der neue Name den letzten Zeichen entsprechen muß (ab dem 17. Buchstaben des neuen Strings). Das bedeutet aber, daß wir uns entweder an die neue Schreibweise für den Infobefehl (z.B. 'ndow', 'ster' oder ähnliches) gewöhnen oder eine Kopie mit normalem Namen auf die Workbench schreiben müssen, was unnötig kostbaren Speicherplatz kostet.

Es geht aber auch einfacher. Wir müssen nur darauf achten, daß das letzte Zeichen des Strings ein Doppelpunkt ist (z.B. Conman-CLI-Window:). Nun weisen wir der Zeichenfolge ow: (von ...Window:) mit Hilfe des ASSIGN-Befehls den INFO-Befehl zu:

ASSIGN ow: c:INFO

Am besten schreiben wir diese Zeile gleich in die Startup-Sequence und speichern diese ab, damit die Anweisung gleich beim Start automatisch ausgeführt wird. Nun wird jedesmal beim Öffnen des CLI (von der Workbench) der Befehl INFO ausgeführt. Auf diese Weise können wir jeden Befehl, der keine weiteren Argumente benötigt, automatisch ausführen lassen.

Die Icons der RamDisks im neuen Design!
Nach dem Booten mit der Workbench, dem Anmelden der RamDisks (RAM: VD0: usw) und dem Starten der Workbench (LoadWB) erscheinen die Icons der RamDisks immer im gleichen Design und an der gleichen Stelle. Man kann sie mit der Maus zwar woanders plazieren, aber nach jedem Neustart ist dies wieder vergessen. Um das zu andern gibt es folgenden Trick:

Zuerst erstellen wir mit einem Icon-Editor (z.B. IconEd) neue Disketten-Icons (Wie das geht steht im AMIGA-Benutzerhandbuch). Dann speichern wir diese Icons auf der Workbenchdiskette zum Beispiel mit den Namen 'ASDGDisk.info', bzw. 'RAMDisk.info' ab.

Aber auf keinen Fall als Disk.info, da sonst das Icon der Workbench überschrieben wird!

In die Startup-Sequence fügt man vor dem Befehl 'LoadWB' folgende zwei Zeilen ein:

```
copy sys:ASDGDisk.info to vd0
:Disk.info
copy sys:RAMDisk.info to ram:
Disk.info
```

Dann wird ein Reset ausgeführt (CTRL, Left-AMIGA, Right-AMIGA) und mit der geänderten Workbench neu gebootet. Nun ist bereits der erste Erfolg sichtbar; die RamDiskIcons erscheinen im neuen Design. Jetzt müssen wir sie nur noch richtig plazieren.

Dazu ziehen wir die Icons mit der Maus an die Stelle, an der sie in Zukunft erscheinen sollen. Nun klicken wir sie mit der Maus einmal an und wählen aus der Menüleiste die Funktion SNAPSHOT. Jetzt ist die Position in der Icon-Struktur gespeichert. Damit diese Einstellung bei jedem Booten sofort wieder zur Verfügung steht, müssen wir die Icons wieder auf die Workbench schreiben.

Dazu öffnen wir ein CLI-Fenster (siehe Handbuch) und geben folgendes ein:

```
copy vd0:Disk.info to sys:ASD
GDisk.info
copy ram:Disk.info to sys:RAM
Disk.info
```

Damit ist alles erledigt. Mit diesem Trick ist es möglich, seiner Workbench ein persönliches Design zu geben.

Star-Druckerzeichensätze nutzen.
Um zum Beispiel mit 'Beckertext' die Druckerzeichensätze des STAR LC 10 (Color) benutzen zu können, genügt es die "direkt ansteuerbaren Druckerbefehle" in den Text einzusetzen.

Um einen Text kursiv, mit Schattendruck (in der Farbe Blau), in doppelter Breite und Höhe mit NLQ und der Schriftart Orator mit Kleinbuchstaben auszugeben, setzt man vor den Text die Zeichenfolge:

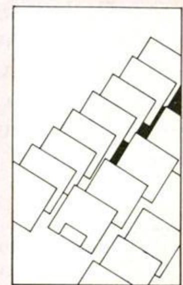
```
"((I))1((B))1((C))2((S))3
((F))3"
```

und mit:

```
"((I))0((B))0((C))0((S))0
((F))9"
```

wird der Drucker wieder auf Normaldruck eingestellt.

(Norbert Dimpfl/jb)



Tausche Tip gegen DATABOX

Haben Ihnen die Kurztips auf unseren Seiten gefallen? Vielleicht haben Sie selbst einen Tip parat, der andere Amigabesitzer interessieren könnte? Dann machen wir Ihnen wieder unser Angebot: Sie schicken einen oder mehrere kurze Tips an uns, und wenn uns diese gefallen, bekommen Sie von uns im Austausch die DATABOX desjenigen Heftes zugeschickt, in dem Ihr Tip erscheint.

Dabei sollten Sie noch folgendes beachten:

Schreiben Sie auf den Brief sowie auf einen kleinen Aufkleber auf der Diskette Ihre lückenlose Anschrift sowie einen kleinen Hinweis, was auf der Diskette zu finden ist.

Dann können Sie sicher sein, daß die DATABOX Sie erreicht. Schicken Sie das Ganze an den

DMV-Verlag
Redaktion AMIGA DOS
Kennwort "Tausche Tip gegen Databox"
Postfach 250
D-3440 Eschwege

Günter Noppeney

Diskettenaufkleber selbst gestalten

Etikettenprinter bringt Ordnung in die Diskettensammlung

Standen Sie auch schon einmal vor dem Problem, aus Ihrem Diskettenarchiv eine bestimmte Diskette herauszusuchen? Aussagekräftige Etiketten erleichtern dabei die Suche ungemein. Mit unserem AmigaBASIC-Programm Etikettenprinter können Sie eigene Diskettenlabels entwerfen und auf dem Drucker ausgeben.

Das Programm Etikettenprint greift auf die dos-, exec- und graphics-Library der Amiga-Extras-Diskette zu. Errichten Sie sich deshalb auf Ihrer Diskette ein Verzeichnis namens bmap und kopieren Sie die genannten Libraries in das bmap-Verzeichnis. Nun können Sie AmigaBASIC laden und das Programm eintippen.

Das Editorfeld: In diesem Feld kann der Cursor mit den Cursortasten gesteuert bzw. mit der Maus positioniert werden. Durch ENTER springt der Cursor an den Anfang der nächsten Reihe.

Ende: Durch Anwählen dieser Taste wird das Programm beendet.

Normal: Normaldruck (24 Zeichen pro Reihe)

Klein: Schmaldruck (43 Zeichen pro Reihe)

Gross: Breitdruck (12 Zeichen pro Reihe)

Reihe: Die Pfeiltasten schieben die Reihe, in der sich der Cursor befindet, in die entsprechende Richtung. Beim Hoch- und Runterschieben kann die Reihe nur bis an den Feldrand geschoben werden. Beim Schieben nach rechts und links wird alles, was über den Feldrand hinausreicht, unweigerlich gelöscht. Die C-Taste löscht nur die Reihe, in der sich der Cursor befindet.

Feld: Die Pfeiltasten schieben das gesamte Feld in die entsprechende Richtung. Alles was über den Feldrand hinausgeschoben wird, wird gelöscht. Die C-Taste löscht das Feld.

Aktuelle Einstellung: Hier wird die aktuelle Schriftart angezeigt, die mit den Schaltern F, K und U eingestellt werden kann. Es sind sieben Kombinationen möglich:

- fett
- kursiv (schräg)
- unterstrichen
- fett und kursiv
- fett und unterstrichen
- kursiv und unterstrichen
- fett, kursiv und unterstrichen

Die Schriftarten werden mit der Maus eingeschaltet und durch nochmaliges Anklicken wieder ausgeschaltet.

Laden: Lädt das ausgewählte Etikett in das Editorfeld.

Speichern: Speichert die Angaben aus dem Editorfeld unter einen definierten Namen. Vorsicht, denn ein bereits existierendes File mit dem gleichen Namen wird ohne Vorwarnung überschrieben.

Löschen: Löscht das angewählte Etikett von der Diskette.

– Sobald eine der drei letztgenannten Optionen aktiviert wird, werden alle Funktionen der unteren Bildschirmhälfte inaktiv. Dafür werden jetzt die vier Schalter rechts oben aktiv. Außerdem erscheinen im Filefeld (rechts oben) alle abgespeicherten Etiketten alphabetisch sortiert. Mit den Pfeiltasten kann man nun diese Liste verschieben, bis das gewünschte Etikett in der hervorgehobenen Reihe steht. Mit der R-Taste kann man den angewählten Modus verlassen.

Im Modus-Speichern befindet sich (in der Liste der Namen) an oberster Stelle eine freie Reihe. Nur dort kann man einen neuen Etikettennamen angeben.

Drucken: Der Inhalt des Editorfeldes wird auf dem Drucker ausgegeben. Das Etikett wird so weit vorgeschoben, daß mit dem Druck auf der Vorderseite der Diskette angefangen wird.

(br)

Wichtig!!!

Die Zeilennummern am Anfang der Programmzeilen dienen nur zur besseren Übersicht und sollten im BASIC-Editor nicht mit eingegeben werden. Sie können zur Eingabe auch den Checksummer (Ausgabe 1/90) benutzen. Die Zeilennummern nicht abtippen, sie dienen lediglich der Übersicht.

Listings

```
000:993 PALETTE 0,.5,.5,.5
001:494 CLEAR,50000&
002:017 DIM b$(13,48):DIM File$(500)
003:681 DIM a%(40):DIM b%(9500):DIM c%(360)
004:108 LIBRARY ":bmap/dos.library"
005:308 LIBRARY ":bmap/exec.library"
006:812 LIBRARY ":bmap/graphics.library"
007:673 DECLARE FUNCTION Lock& LIBRARY
008:425 DECLARE FUNCTION ExNext LIBRARY
009:609 DECLARE FUNCTION Examine LIBRARY
010:049 DECLARE FUNCTION AllocMem& LIBRARY
011:129 DECLARE FUNCTION AskSoftStyle% LIBRARY
012:945 DECLARE FUNCTION SetSoftStyle% LIBRARY
013:000
014:832 c=24:k=3:l=1:m=1:o=1:p=1:q=1:t=0:z=0:g&=5000:r&=2^16
015:097 GET (0,0)-(43*8+5,104),b%
016:476 WINDOW 1,"            Etikettenprinter
                  By G.Noppeney in 1989      ",(0,0)-(630,18
        5),0
017:897 ON MOUSE GOSUB Maus
018:992 MOUSE ON
019:000
020:477 Info&=AllocMem&(g&,r&)
021:322 GOSUB GetVerz
022:000
023:074 Start:
024:223   GOSUB Neu
025:000
026:096 Abfrage:
027:936   IF t=1 THEN
028:488     IF (v=2 AND w=5) THEN
029:826       a$=INKEY$
030:632       IF a$="" AND w5=1 THEN w5=0:w1=0:NeuFile$="":S
        CROLL (408,53)-(568,60),0,8
031:823 c:    IF a$="" THEN Abfrage
032:079       IF ASC(a$)=  8 THEN w1=w1-1:l1=1:a$=" ":NeuFil
        e$=LEFT$(NeuFile$,LEN(NeuFile$)-1)
033:639       IF w1=20 THEN Abfrage
034:847       IF ASC(a$)= 13 THEN Abfrage
```

Listing Etikettenprint

```
035:597       LOCATE 8,52+w1:PRINT a$:SCROLL (408+w1*8,53)-(
        415+w1*8,63),0,-3
036:542       LINE (404,62)-(572,62),2
037:583       IF l1=0 THEN NeuFile$=NeuFile$+a$:w1=w1+1
038:082       l1=0
039:216     END IF .
040:287     GOTO Abfrage
041:216   END IF
042:364   LOCATE y,x
043:432   GOSUB Feld
044:188 a:
045:826   a$=INKEY$
046:169   IF a$="" THEN a
047:432   GOSUB Feld
048:715   IF t=1 AND w=5 AND v=2 THEN c
049:327   IF t=1 THEN Abfrage
050:000
051:654   IF ASC(a$)=8 THEN x=x-1:a$=" ":a=1:GOSUB Test
052:343   IF ASC(a$)=13 THEN x=6:y=y+1:GOSUB Test:GOTO Abfra
        ge
053:183   IF ASC(a$)=28 THEN y=y-1:GOSUB Test:GOTO Abfrage
054:615   IF ASC(a$)=29 THEN y=y+1:GOSUB Test:GOTO Abfrage
055:999   IF ASC(a$)=30 THEN x=x+1:GOSUB Test:GOTO Abfrage
056:863   IF ASC(a$)=31 THEN x=x-1:GOSUB Test:GOTO Abfrage
057:000
058:354   IF a=1 THEN Stil 0:e$=d$:d$="0"
059:368   b$(y,x)=a$+d$
060:364   LOCATE y,x
061:135   PRINT a$;
062:720   IF a=1 THEN
063:838     d$=e$:x=x-1:a=0
064:109     IF d$="1" THEN Stil 1
065:134     IF d$="2" THEN Stil 2
066:159     IF d$="3" THEN Stil 3
067:184     IF d$="4" THEN Stil 4
068:209     IF d$="5" THEN Stil 5
069:234     IF d$="6" THEN Stil 6
070:259     IF d$="7" THEN Stil 7
071:216   END IF
072:383   x=x+1
073:891   IF x>(c+5) THEN x= 6:y=y+1
```

Listing Etikettenprint

```
074:256   IF y>13 THEN y= 2
075:287 GOTO Abfrage
076:000
077:170 Feld:
078:167   GET (x*8-8,y*8-8)-(x*8-6,y*8-1),a%
079:866   PUT (x*8-8,y*8-8),a%,PRESET
080:982 RETURN
081:000
082:316 Maus:
083:073   a1=MOUSE(0)
084:635   x1=MOUSE(5)
085:661   y1=MOUSE(6)
086:000
087:936   IF t=1 THEN
088:704     IF x1>589 AND x1<613 THEN
089:663       IF y1>34 AND y1<46 THEN Hoch3
090:785       IF y1>50 AND y1<62 THEN OK
091:731       IF y1>66 AND y1<78 THEN Runter3
092:509       IF y1>90 AND y1<102 THEN v=0:GOTO OK
093:216     END IF
094:982     RETURN
095:216   END IF
096:000
097:408   IF x1>39 AND x1<(c*8+41) THEN
098:360     IF y1>7 AND y1<105 THEN
099:432       GOSUB Feld
100:387       x=INT(x1/8)+1
101:547       y=INT(y1/8)+1
102:364       LOCATE y,x
103:432       GOSUB Feld
104:216     END IF
105:216   END IF
106:000
107:024   IF y1>117 AND y1<129 THEN
108:364     IF x1>17 AND x1<77 THEN CLS:CALL FreeMem(Info&,g
&):LIBRARY CLOSE:END
109:844     IF x1>531 AND x1<611 THEN v=1:GOSUB Anzeigen
110:216   END IF
111:000
112:024   IF y1>133 AND y1<145 THEN
113:654     IF x1>17 AND x1<77 THEN c=24:k=3:l=1:m=1:e=1:GOS
UB Neu:e=0
114:565     IF x1>137 AND x1<159 THEN GOSUB Hoch1
115:430     IF x1>249 AND x1<271 THEN GOSUB Hoch2
116:447     IF x1>331 AND x1<363 THEN r=1:o=o+2:IF o=5 THEN
o=1
117:228     IF x1>531 AND x1<611 THEN v=2:GOSUB Anzeigen
```

Listing Etikettenprint

```
118:216   END IF
119:000
120:560   IF y1>149 AND y1<161 THEN
121:830     IF x1>17 AND x1<77 THEN c=43:k=1:l=3:m=1:e=1:GOS
UB Neu:e=0
122:729     IF x1>107 AND x1<129 THEN GOSUB Ziehen1
123:351     IF x1>137 AND x1<159 THEN GOSUB RNeu
124:233     IF x1>167 AND x1<189 THEN GOSUB Schieben1
125:186     IF x1>219 AND x1<241 THEN GOSUB Ziehen2
126:518     IF x1>249 AND x1<271 THEN e=1:GOSUB Neu:e=0
127:754     IF x1>279 AND x1<301 THEN GOSUB Schieben2
128:803     IF x1>331 AND x1<363 THEN r=1:p=p+2:IF p=5 THEN
p=1
129:508     IF x1>531 AND x1<611 THEN v=1::vv=1:GOSUB Anzeig
en
130:216   END IF
131:000
132:688   IF y1>165 AND y1<177 THEN
133:286     IF x1>17 AND x1<77 THEN c=12:k=1:l=1:m=3:e=1:GOS
UB Neu:e=0
134:657     IF x1>137 AND x1<159 THEN GOSUB Runter1
135:114     IF x1>249 AND x1<271 THEN GOSUB Runter2
136:159     IF x1>331 AND x1<363 THEN r=1:q=q+2:IF q=5 THEN
q=1
137:132     IF x1>531 AND x1<611 THEN GOSUB Drucken
138:216   END IF
139:000
140:060   IF r=1 THEN r=0:GOSUB Schrift
141:982 RETURN
142:000
143:454 Test:
144:175   IF x< 6 THEN x=c+5:y=y-1
145:945   IF y< 2 THEN y=13
146:891   IF x>(c+5) THEN x= 6:y=y+1
147:256   IF y>13 THEN y= 2
148:982 RETURN
149:000
150:834 Drucken:
151:154   COLOR 3:LOCATE 22,69:PRINT "Drucken"
152:955   OPEN "prt:" FOR OUTPUT AS 1
153:491   PRINT #1,CHR$(27);"c";
154:371   PRINT #1,CHR$(27);"E";
155:371   PRINT #1,CHR$(27);"E";
156:371   PRINT #1,CHR$(27);"E";
157:037   PRINT #1,CHR$(27)"[0w";CHR$(27)"[5w";CHR$(18);
158:569   IF c=43 THEN PRINT #1,CHR$(15);
```

Listing Etikettenprint

```
159:363    IF c=12 THEN PRINT #1,CHR$(27)"[6w";
160:000
161:967    i=-1
162:609    FOR a=2 TO 13
163:583     FOR b=6 TO c+5
164:557      a$=RIGHT$(b$(a,b),1):h=VAL(a$)
165:792      IF i<>h THEN
166:243        IF (i AND 1)=1 AND (h AND 1)<>1 THEN PRINT #1,
CHR$(27)+"[24m";
167:947        IF (i AND 2)=2 AND (h AND 2)<>2 THEN PRINT #1,
CHR$(27)+"[22m";
168:395        IF (i AND 4)=4 AND (h AND 4)<>4 THEN PRINT #1,
CHR$(27)+"[23m";
169:323        IF (h AND 4)=4 AND (i AND 4)<>4 THEN PRINT #1,
CHR$(27)+"[3m";
170:587        IF (h AND 2)=2 AND (i AND 2)<>2 THEN PRINT #1,
CHR$(27)+"[1m";
171:251        IF (h AND 1)=1 AND (i AND 1)<>1 THEN PRINT #1,
CHR$(27)+"[4m";
172:486        i=h
173:216      END IF
174:833      PRINT #1,LEFT$(b$(a,b),1);
175:160     NEXT
176:115     PRINT #1,CHR$(27)+"E";
177:160    NEXT
178:511    CLOSE 1
179:698    COLOR 1:LOCATE 22,69:PRINT "Drucken"
180:982 RETURN
181:000
182:197 SUB Stil (nr%) STATIC
183:627    bits% = AskSoftStyle%(WINDOW(8))
184:639    news% = SetSoftStyle%(WINDOW(8),nr%,bits%)
185:016    IF (nr% AND 4) = 4 THEN
186:369      CALL SetDrMd(WINDOW(8),0)
187:091    ELSE
188:371      CALL SetDrMd(WINDOW(8),1)
189:216    END IF
190:568 END SUB
191:000
192:440 RNeu:
193:256    FOR b=6 TO c+5:b$(y,b)=" 0":NEXT
194:815    y2=y*8-8:y3=y*8-1:dx=0:dy=1
195:464 GOTO Scrollen
196:000
197:820 Schieben1:
198:408    FOR b=c+5 TO 7 STEP -1:b$(y,b)=b$(y,b-1):NEXT
199:182    b$(y,6)=" 0":y2=y*8-8:y3=y*8-1:dx=1:dy=0
200:464 GOTO Scrollen
201:000
202:572 Ziehen1:
203:696    FOR b=6 TO c+4:b$(y,b)=b$(y,b+1):NEXT
204:590    b$(y,c+5)=" 0":y2=y*8-8:y3=y*8-1:dx=-1:dy=0
205:464 GOTO Scrollen
206:000
207:548 Hoch1:
208:390    IF y=2 THEN RETURN
209:448    FOR b=6 TO c+5:b$(y-1,b)=b$(y,b):b$(y,b)=" 0":NEXT
210:823    y2=y*8-16:y3=y*8-1:dx=0:dy=-1:j=-1
211:464 GOTO Scrollen
212:000
213:812 Runter1:
214:710    IF y=13 THEN RETURN
215:800    FOR b=6 TO c+5:b$(y+1,b)=b$(y,b):b$(y,b)=" 0":NEXT
216:739    y2=y*8-8:y3=y*8+7:dx=0:dy=1:j=1
217:464 GOTO Scrollen
218:000
219:822 Schieben2:
220:441    FOR n=2 TO 13
221:739      FOR b=c+5 TO 7 STEP -1
222:079        b$(n,b)=b$(n,b-1)
223:160      NEXT
224:386      b$(n,6)=" 0"
225:160    NEXT
226:286    y2=8:y3=103:dx=1:dy=0
227:464 GOTO Scrollen
228:000
229:574 Ziehen2:
230:441    FOR n=2 TO 13
231:582      FOR b=6 TO c+4
232:071        b$(n,b)=b$(n,b+1)
233:160      NEXT
234:314      b$(n,c+5)=" 0"
235:160    NEXT
236:886    y2=8:y3=103:dx=-1:dy=0
237:464 GOTO Scrollen
238:000
239:550 Hoch2:
240:457    FOR n=3 TO 13
241:583      FOR b=6 TO c+5
242:173        b$(n-1,b)=b$(n,b)
243:160      NEXT
244:160    NEXT
245:288    FOR b=6 TO c+5:b$(13,b)=" 0":NEXT
246:551    y2=8:y3=103:dx=0:dy=-1
247:464 GOTO Scrollen
248:000
249:814 Runter2:
250:859    FOR n=12 TO 2 STEP -1
251:583      FOR b=6 TO c+5
252:981        b$(n+1,b)=b$(n,b)
253:160      NEXT
254:160    NEXT
255:768    FOR b=6 TO c+5:b$(2,b)=" 0":NEXT
256:255    y2=8:y3=103:dx=0:dy=1
257:464 GOTO Scrollen
258:000
```

Listing Etikettenprint

```
259:914 Scrollen:
260:432    GOSUB Feld
261:140    x2=40:x3=c*8+40
262:872    FOR n=1 TO 8:SCROLL (x2,y2)-(x3,y3),dx,dy:NEXT
263:778    y=y+j:j=0
264:432    GOSUB Feld
265:982 RETURN
266:000
267:128 Neu:
268:928    IF e<>2 THEN
269:492      Stil 0
270:609      FOR a=2 TO 13
271:583        FOR b=6 TO c+5
272:826          b$(a,b)=" 0"
273:160        NEXT
274:160      NEXT
275:216    END IF
276:000
277:378 PUT (39,7),b%,PSET
278:130 LINE (c*8+41,7)-(c*8+42,105),2,b:LINE (c*8+43,7)-(c*
8+44,106),2,b
279:581 LINE (38,7)-(39,106),1,b:LINE (39,7)-(c*8+42,7),1
280:998 LINE (39,105)-(c*8+41,105),2:LINE (38,106)-(o*8+42,1
06),2
281:654 COLOR k:LOCATE 18,4:PRINT "Normal"
282:242 COLOR 1:LOCATE 20,4:PRINT "Klein "
283:942 COLOR m:LOCATE 22,4:PRINT "Gross "
284:817 x=6:y=2:a=0:d$="0":COLOR 1
285:000
286:992 IF e>0 THEN
287:432    GOSUB Feld
288:982    RETURN
289:216 END IF
290:000
291:106 LINE (617,114)-(618,180),2,b:LINE (619,114)-(620,181
),2,b
292:758 LINE (10,114)-(11,180),1,b:LINE (11,114)-(618,114),1
,b
293:086 LINE (11,180)-(617,180),2:LINE (10,181)-(618,181),2
294:000
295:822 LOCATE 16,4:PRINT " Ende "
296:542 LINE (17,117)-(77,129),1,b:LINE (77,117)-(78,129),2,
b:LINE (17,129)-(77,129),2
297:542 LINE (17,133)-(77,145),1,b:LINE (77,133)-(78,145),2,
b:LINE (17,145)-(77,145),2
298:022 LINE (17,149)-(77,161),1,b:LINE (77,149)-(78,161),2,
b:LINE (17,161)-(77,161),2
299:894 LINE (17,165)-(77,177),1,b:LINE (77,165)-(78,177),2,
b:LINE (17,177)-(77,177),2
300:996 LOCATE 16,17:PRINT"Reihe"
301:206 LINE (121,117)-(173,129),1,b:LINE (120,117)-(121,129
),2,b:LINE (121,117)-(173,117),2
302:856 LOCATE 16,31:PRINT"Feld":SCROLL (234,118)-(284,128),
4,0
303:006 LINE (233,117)-(285,129),1,b:LINE (232,117)-(233,129
),2,b:LINE (233,117)-(285,117),2
304:462 LINE (331,117)-(515,129),1,b:LINE (330,117)-(331,129
),2,b:LINE (331,117)-(515,117),2
305:000
306:902 LINE (107,149)-(129,161),1,b:LINE (128,149)-(129,161
),2,b:LINE (107,161)-(129,161),2
307:390 LINE (137,133)-(159,145),1,b:LINE (158,133)-(159,145
),2,b:LINE (137,145)-(159,145),2
308:502 LINE (137,149)-(159,161),1,b:LINE (158,149)-(159,161
),2,b:LINE (137,161)-(159,161),2
309:190 LINE (137,165)-(159,177),1,b:LINE (158,165)-(159,177
),2,b:LINE (137,177)-(159,177),2
310:102 LINE (167,149)-(189,161),1,b:LINE (188,149)-(189,161
),2,b:LINE (167,161)-(189,161),2
311:573 LINE (111,155)-(125,155),1:LINE (111,155)-(116,152),
1:LINE (111,155)-(116,158),1
312:237 LINE (148,136)-(148,142),1:LINE (148,136)-(143,139),
1:LINE (148,136)-(153,139),1
313:317 LINE (148,168)-(148,174),1:LINE (148,174)-(143,171),
1:LINE (148,174)-(153,171),1
314:773 LINE (171,155)-(185,155),1:LINE (185,155)-(180,152),
1:LINE (185,155)-(180,158),1
315:512 LOCATE 20,19:PRINT"C"
316:000
317:614 LINE (219,149)-(241,161),1,b:LINE (240,149)-(241,161
),2,b:LINE (219,161)-(241,161),2
318:102 LINE (249,133)-(271,145),1,b:LINE (270,133)-(271,145
),2,b:LINE (249,145)-(271,145),2
319:214 LINE (249,149)-(271,161),1,b:LINE (270,149)-(271,161
),2,b:LINE (249,161)-(271,161),2
320:902 LINE (249,165)-(271,177),1,b:LINE (270,165)-(271,177
),2,b:LINE (249,177)-(271,177),2
321:638 LINE (279,149)-(301,161),1,b:LINE (300,149)-(301,161
),2,b:LINE (279,161)-(301,161),2
322:173 LINE (223,155)-(237,155),1:LINE (223,155)-(228,152),
1:LINE (223,155)-(228,158),1
323:997 LINE (260,136)-(260,142),1:LINE (260,136)-(255,139),
1:LINE (260,136)-(265,139),1
324:077 LINE (260,168)-(260,174),1:LINE (260,174)-(255,171),
1:LINE (260,174)-(265,171),1
325:653 LINE (283,155)-(297,155),1:LINE (297,155)-(291,152),
1:LINE (297,155)-(291,158),1
326:488 LOCATE 20,33:PRINT"C"
327:000
328:788 GOSUB Schrift
329:000
330:106 LOCATE 18,48:Stil 2:PRINT "Fett"
331:890 LOCATE 20,48:Stil 4:PRINT "Kursiv"
332:186 LOCATE 22,48:Stil 1:PRINT "Unterstrichen"
333:492 Stil 0
334:666 LOCATE 16,70:PRINT "Laden"
335:334 LOCATE 18,68:PRINT "Speichern"
```

Listing Etikettenprint

```
336:922 LOCATE 20,69:PRINT "Löschen"
337:826 LOCATE 22,69:PRINT "Drucken"
338:574 LINE (531,117)-(611,129),1,b:LINE (610,117)-(611,129
        ),2,b:LINE (531,129)-(611,129),2
339:574 LINE (531,133)-(611,145),1,b:LINE (610,133)-(611,145
        ),2,b:LINE (531,145)-(611,145),2
340:686 LINE (531,149)-(611,161),1,b:LINE (610,149)-(611,161
        ),2,b:LINE (531,161)-(611,161),2
341:000
342:374 LINE (531,165)-(611,177),1,b:LINE (610,165)-(611,177
        ),2,b:LINE (531,177)-(611,177),2
343:517 LOCATE 8,75:PRINT "OK":SCROLL (589,50)-(613,62),2,-3
344:173 LINE (601,36)-(601,44),1:LINE (601,36)-(596,38),1:LI
        NE (601,36)-(606,38),1
345:565 LINE (601,68)-(601,76),1:LINE (601,76)-(596,74),1:LI
        NE (601,76)-(606,74),1
346:793 LOCATE 12,75:PRINT "R":SCROLL (589,88)-(613,99),5,5
347:750 LINE (577,7)-(578,105),2,b:LINE (579,7)-(579,106),2
348:845 LINE (398,7)-(399,105),1,b:LINE (398,7)-(578,7),1
349:022 LINE (399,105)-(577,105),2:LINE (398,106)-(578,106),
        2
350:658 LINE (585,7)-(617,105),1,b:LINE (616,7)-(617,105),2,
        b:LINE (585,105)-(617,106),2,b
351:000
352:734 LINE (589,34)-(613,46),1,b:LINE (612,34)-(613,46),2,
        b:LINE (589,46)-(613,46),2
353:734 LINE (589,50)-(613,62),1,b:LINE (612,50)-(613,62),2,
        b:LINE (589,62)-(613,62),2
354:086 LINE (589,66)-(613,78),1,b:LINE (612,66)-(613,78),2,
        b:LINE (589,78)-(613,78),2
355:926 LINE (404,50)-(572,62),1,b:LINE (571,50)-(572,62),2,
        b:LINE (404,62)-(572,62),2
356:046 LINE (589,90)-(613,102),1,b:LINE (612,90)-(613,102),
        2,b:LINE (589,102)-(613,102),2
357:000
358:862 LINE (331,133)-(363,145),1,b:LINE (362,133)-(363,145
        ),2,b:LINE (331,145)-(363,145),2
359:974 LINE (331,149)-(363,161),1,b:LINE (362,149)-(363,161
        ),2,b:LINE (331,161)-(363,161),2
360:662 LINE (331,165)-(363,177),1,b:LINE (362,165)-(363,177
        ),2,b:LINE (331,177)-(363,177),2
361:982 RETURN
362:000
363:618 Schrift:
364:492 Stil 0
365:414 COLOR o:LOCATE 18,44:PRINT "F"
366:000 COLOR p:LOCATE 20,44:PRINT "K"
367:364 COLOR q:LOCATE 22,44:PRINT "U"
368:545 COLOR 1
369:082 LOCATE 16,43:PRINT "                              "
370:844 IF o=1 AND p=1 AND q=1 THEN d$="0":a$="":Stil 0
371:598 IF o=3 AND p=1 AND q=1 THEN d$="2":a$="":Stil 2
372:528 IF o=1 AND p=3 AND q=1 THEN d$="4":a$="":Stil 4
373:645 IF o=1 AND p=1 AND q=3 THEN d$="1":a$="":Stil 1
374:282 IF o=3 AND p=3 AND q=1 THEN d$="6":a$="":Stil 6
375:399 IF o=3 AND p=1 AND q=3 THEN d$="3":a$="":Stil 3
376:329 IF o=1 AND p=3 AND q=3 THEN d$="5":a$="":Stil 5
377:083 IF o=3 AND p=3 AND q=3 THEN d$="7":a$="":Stil 7
378:224 LOCATE 16,44:PRINT "Aktuelle Einstellung"
379:982 RETURN
380:000
381:818 Anzeigen:
382:988   t=1:Stil 0:o=1:p=1:q=1:w1=0:NeuFile$="":GOSUB Schr
        ift
383:946   IF v=1 AND vv=0 THEN COLOR 3:LOCATE 16,70:PRINT "L
        aden"
384:514   IF v=1 AND vv=1 THEN COLOR 3:LOCATE 20,69:PRINT "L
        öschen"
385:086   IF v=2 THEN COLOR 3:LOCATE 18,68:PRINT "Speichern"
386:545   COLOR 1
387:456   FOR n=6 TO 11-v:LOCATE n+v+2,52:PRINT File$(n):NEX
        T
388:272   w=6-v
389:999   GOSUB Hoch3
390:401   LOCATE 13,52:PRINT File$(12-v)
391:317   IF z=0 AND v<2 THEN v=0:GOTO OK
392:982 RETURN
393:000
394:552 Hoch3:
395:574   IF w=(z+5) THEN RETURN
396:000   IF w=5 THEN SCROLL (408,53)-(568,60),0,8
397:506   SCROLL (408,8)-(568,47),0,-8
398:043   GET (408,53)-(568,60),c%
399:378   PUT (408,40),c%,PSET
400:667   GET (408,64)-(568,71),c%
401:938   PUT (408,53),c%,PSET
402:834   SCROLL (408,64)-(568,103),0,-8
403:363   w=w+1
404:927   LOCATE 13,52:PRINT File$(w+5)
405:982 RETURN
406:000
407:816 Runter3:
408:878   IF (w-7+v)=0 THEN RETURN
409:400   SCROLL (408,64)-(568,103),0,8
410:043   GET (408,53)-(568,60),c%
411:474   PUT (408,64),c%,PSET
412:131   GET (408,40)-(568,47),c%
413:938   PUT (408,53),c%,PSET
414:736   SCROLL (408,8)-(568,47),0,8
415:215   w=w-1:IF w=5 THEN w5=1
416:519   LOCATE 2,52:PRINT File$(w-5)
417:982 RETURN
418:000
419:524 OK:
420:352   IF v=2 AND w=5 THEN
421:380     w=z+6
422:390     File$(w)=NeuFile$
423:216   END IF
```

Listing Etikettenprint

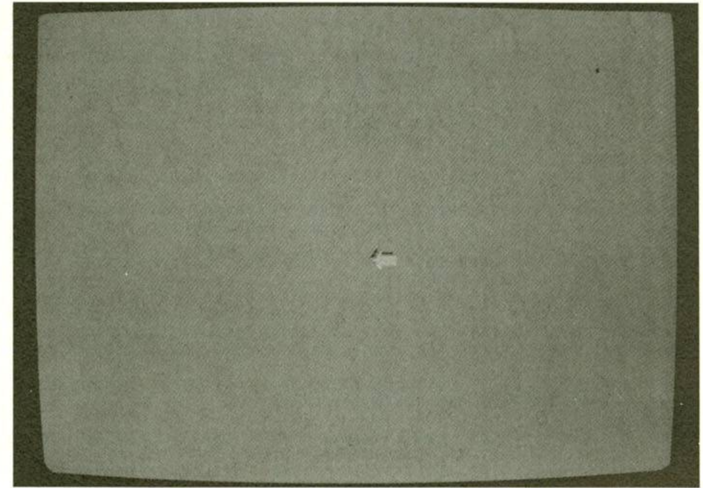
```
424:400   IF v=1 AND vv=0 THEN GOSUB Laden
425:840   IF v=1 AND vv=1 THEN GOSUB Loeschen
426:158   IF v=2 THEN GOSUB Speichern
427:328   FOR n=2 TO 6:LOCATE n,52:PRINT "
          ":NEXT
428:176   FOR n=9 TO 13:LOCATE n,52:PRINT "
          ":NEXT
429:131   GET (408,40)-(568,47),c%
430:938   PUT (408,53),c%,PSET
431:432   GOSUB Feld
432:322   t=0:vv=0
433:545   COLOR 1
434:922   LOCATE 20,69:PRINT "Löschen"
435:334   LOCATE 18,68:PRINT "Speichern"
436:666   LOCATE 16,70:PRINT "Laden"
437:982 RETURN
438:000
439:506 Laden:
440:086   SourceFile$=Verz$+"/"+File$(w)+".ET"
441:547   OPEN SourceFile$ FOR INPUT AS 1
442:597   INPUT#1,c,k,l,m
443:609   FOR a=2 TO 13
444:583     FOR b=6 TO c+5
445:765       INPUT#1,b$(a,b)
446:160     NEXT
447:160   NEXT
448:511   CLOSE 1
449:062   e=2:GOSUB Neu:e=0
450:609   FOR a=2 TO 13
451:583     FOR b=6 TO c+5
452:501       IF LEN(b$(a,b))=1 THEN b$(a,b)=" "+b$(a,b)
453:771       c$=d$:d$=RIGHT$(b$(a,b),1)
454:280       IF c$<>d$ THEN
455:084         IF d$="0" THEN Stil 0
456:109         IF d$="1" THEN Stil 1
457:134         IF d$="2" THEN Stil 2
458:159         IF d$="3" THEN Stil 3
459:184         IF d$="4" THEN Stil 4
460:209         IF d$="5" THEN Stil 5
461:234         IF d$="6" THEN Stil 6
462:259         IF d$="7" THEN Stil 7
463:216       END IF
464:857       LOCATE a,b:PRINT LEFT$(b$(a,b),1);
465:160     NEXT
466:160   NEXT
467:982 RETURN
468:000
469:870 Speichern:
470:822   DestFile$=Verz$+"/"+File$(w)+".ET"
471:835   OPEN DestFile$ FOR OUTPUT AS 1
472:045   PRINT #1,c,k,l,m
473:609   FOR a=2 TO 13
474:583     FOR b=6 TO c+5
475:213       PRINT#1,b$(a,b)
476:160     NEXT
477:160   NEXT
478:511   CLOSE 1
479:716   IF w=z+6 THEN z=z+1:GOSUB Sort
480:982 RETURN
481:000
482:778 Loeschen:
483:086   SourceFile$=Verz$+"/"+File$(w)+".ET"
484:134   KILL SourceFile$
485:232   FOR a=w TO z+5:File$(a)=File$(a+1):NEXT
486:666   z=z-1:GOSUB Sort:vv=0
487:982 RETURN
488:000
489:686 GetVerz:
490:812   Verz$ = ":Disks"
491:513   Ver$  = Verz$ + CHR$(0)
492:077   VerzL& = Lock&(SADD(Ver$), -2&)
493:937   Status = Examine(VerzL&, Info&)
494:781   CALL Ausgabe(Info&,"")
495:598   Verzeichnis$=File$
496:697   Status = ExNext(VerzL&, Info&)
497:657   WHILE(Status<>0)
498:357     CALL Ausgabe(Info&,Verzeichnis$)
499:697     Status = ExNext(VerzL&, Info&)
500:196   WEND
501:511   CLOSE 1
502:000
503:514 Sort:
504:378   FOR ii = 6 TO z + 4
505:263     kk = ii
506:603     FOR jj = ii TO z + 5
507:706       IF File$(jj)<File$(kk) THEN kk = jj
508:862     NEXT jj
509:251     SWAP File$(ii), File$(kk)
510:859   NEXT ii
511:752   FOR ii=0 TO 5:File$(ii)="":NEXT
512:808   FOR ii=z+6 TO z+11:File$(ii)="":NEXT
513:982 RETURN
514:000
515:053 SUB Ausgabe(Mem&,Ve-$) STATIC
516:918   SHARED File$(),z
517:298   ii = 8 : File$=""
518:289   WHILE(PEEK(Mem&+ii)<>0)
519:559     File$ = File$ + CHR$(PEEK(Mem&+ii))
520:995     ii = ii + 1
521:196   WEND
522:696   IF RIGHT$(File$,3) = ".ET" THEN
523:579     File$(6+z)=LEFT$(File$,LEN(File$)-3)
524:423     z=z+1
525:640     EXIT SUB
526:216   END IF
527:568 END SUB
```

Listing Etikettenprint

Ulf Peterson

SPRITES

Den ehemaligen C-64-Usern dürfte die Bedeutung der Sprites bereits bekannt sein. Sollten Sie aber nicht zu diesem Personenkreis gehören, so sei zur Einführung erst einmal gesagt, daß es sich bei den Sprites um Objekte handelt, die sich, ganz im Gegensatz zu den sogenannten BOBs (Blitter Objects), völlig unabhängig von den Bitplanes über den Hintergrund bewegen lassen. Im Ganzen ist der Amiga in der Lage, bis zu acht solcher Sprites gleichzeitig auf dem Bildschirm darzustellen, wobei ein Sprite maximal zwar nur 16 Punkte breit, dafür aber vertikal der Länge des Bildschirms entsprechend lang sein kann. An Farben sind bei den Sprites bis zu vier pro Objekt verfügbar. Dabei entspricht die vierte Farbe der Hintergrundfarbe ($dff180), das heißt, daß an den Stellen, wo diese Farbe angewählt wird, das Sprite transparent ist. Im Verlauf dieses Artikels wird aber noch gezeigt, wie es durch einen kleinen Trick möglich ist, Sprites mit insgesamt 16 Farben darzustellen.



Um die Sprites verschiedenfarbig erscheinen zu lassen, sind zwei sogenannte Mini-Bitplanes vonnöten, die genau wie ihre großen Kollegen aufgebaut sind.

Aufbau der Sprites

Diese Mini-Bitplanes liegen jedoch nicht wie bei ihren Brüdern hintereinander im Speicher, sondern vielmehr nebeneinander vor. Wie oben bereits beschrieben, liegt die maximale Breite eines Sprites bei 16 Punkten (Datenwort), was exakt 16 Pixeln im LoRes-Modus entsprechen würde. Um nun jeden dieser Punkte in einer von vier möglichen Farben erscheinen zu lassen, sind genau zwei Bits nötig, die insgesamt als Kombination vier verschiedene Ergebnisse liefern können. Der Aufbau des Grafikbereichs einer Spritedatenliste sieht so aus, daß mit Hilfe von sogenannten Langworten jeweils die beiden ersten und folgenden Zeilen beider Bitplanes als zusammengefaßte Datenwörter in eben dieser Liste hintereinander vorliegen. In den ersten 16 Bits eines dieser Langwörter befindet sich die eine Zeile der ersten Bitplane und in der anderen Hälfte die gleiche Zeile der zweiten Bit plane. Dazu zum besseren Verständnis erst einmal ein Beispiel. Stellen Sie sich folgende Zeilen als ein kleines Sprite vor. Auf den ersten Blick mag das etwas schwierig wirken, aber versuchen Sie es einmal. Die verwendeten Zahlen stehen für die entsprechende Spritefarbe (0 -> Hintergrundfarbe).

Tabelle 1 zeigt die Zusammenhänge. Rechts neben dem Sprite erkennen Sie die einzelnen Zeilen, zerlegt in Form von Datenwörtern der beiden Bitplanes, die beide in einem Langwort bereits untergebracht wurden. Würde man die beiden Datenwörter der einzelnen Zeilen wieder zusammenfassen, so ergäben deren Kombinationen die entsprechenden Farben der einzelnen Punkte des Sprites. Beispielsweise würden das Bit 31 (erste Bitplane) und das Bit 15 (zweite Bitplane) aus der ersten Zeile die Farbe des in der linken oberen Ecke liegenden Punktes beim Sprite ergeben.

Farbzuweisung

Einer der gravierendsten Nachteile der Sprites ist darin zu finden, daß sie keine eigene Farbtabelle besitzen, sondern die hintere Tabellenhälfte der Grafikfarben mitbenutzen ($dff1a0 bis $dff1be). Wenn man also mehr als vier Bitplanes für seine Grafik verwendet, kann es durchaus passieren, daß die Sprites sprichwörtlich in der Grafik untergehen oder einfach nur noch sehr schlecht zu erkennen sind. Noch trauriger ist meiner Meinung nach aber die Tatsache, daß immer zwei Sprites dieselben Farbregister benutzen. Im Grunde genommen existieren also nur vier verschiedenfarbige Sprites. Vielleicht verstehen Sie nun auch etwas besser, warum bei vielen Spielen nicht von Sprites, sondern lieber von BOBs Gebrauch gemacht wird. Zwar könnte man die Sprites mit Hilfe des Coppers farbiger gestalten, jedoch lohnt sich der Aufwand meistens nicht.

Tabelle 2 zeigt die Farbtabelle für die Sprites.

Spritestruktur

Kommen wir nun an dieser Stelle zum Aufbau einer sogenannten Spritedatenliste. Wie so oft beim Amiga handelt es sich hier wiederum um eine Struktur, an deren Anfang zwei Kontrollwörter und an deren Ende entweder wieder zwei Kontrollwörter oder eine Endmarkierung stehen. Auf diese Weise ist es möglich, auf einem Sprite-DMA-Kanal mehrere Sprites darstellen zu können, ohne überflüssige Nullbytes verwenden zu müssen, wenn zum Beispiel ein Sprite auf der oberen Bildschirmhälfte und eins auf der unteren Hälfte dargestellt

Sprite	Listenlangwort	
	Bits 31 – 16	Bits 15 – 0
0000111111110000	0000111111110000	0000000000000000
0001122222211000	0001100000011000	0000011111100000
0001133333311000	0001111111111000	0000011111100000
0000111111110000	0000111111110000	0000000000000000

Tabelle 1

FARBTABELLE FÜR DIE SPRITES: DW1 = Bits 31-16 DW0 = Bits 15-0

Sprite	DW1	DW0	Register
0 und 1	0	0	transparent
"	1	0	$dff1a2
"	0	1	$dff1a4
"	1	1	$dff1a6
2 und 3	0	0	transparent
"	1	0	$dff1aa
"	0	1	$dff1ac
"	1	1	$dff1ae
4 und 5	0	0	transparent
"	1	0	$dff1b2
"	0	1	$dff1b4
"	1	1	$dff1b6
6 und 7	0	0	transparent
"	1	0	$dff1ba
"	0	1	$dff1bc
"	1	1	$dff1be

Tabelle 2

werden sollen. Tabelle 3 zeigt das Aussehen der Kontrollwörter.
Sowohl für die horizontale als auch für die vertikale Position sind jeweils neun Bits vorgesehen, wobei man normalerweise jedoch nur die "normalen" 8 Bit benötigt. Wie bereits oben schon beschrieben, beträgt die Auflösung in horizontaler Richtung einen niedrigauflösenden Pixel und in vertikaler Richtung eine Rasterzeile. Es muß übrigens darauf geachtet werden, daß die Positionswerte innerhalb der Werte von DIWSTRT und DIWSTOP, also innerhalb des Bildschirmfensters, liegen, da ansonsten keine (!) Spritedarstellung erfolgen kann. Die horizontale und die vertikale Startposition beziehen sich auf die linke obere Ecke des betroffenen Sprites. Bitte beachten Sie unbedingt, daß die eigentliche Endzeile des Sprites um eins erhöht im zweiten Kontrollwort eingetragen werden muß.
Tabelle 4 zeigt das Beispiel einer Datenliste.
Es gilt übrigens zu beachten, daß immer die Hälfte von HANF in das erste Kontrollwort eingetragen wird. Der Grund hierfür ist darin zu suchen, daß ein Spritepunkt einem LoRes-Pixel entspricht.
Wollte man auf dem gleichen DMA-Spritekanal ein weiteres Sprite ab einer tieferen Vertikalposition darstellen, sind bloß anstelle der Endmarkierung zwei weitere Kontrollwörter zu schreiben. Dabei ist allerdings zu beachten, daß die neue vertikale Anfangsposition um mindestens eins höher sein muß als die vorige Endposition VEND, da der DMA eine Zeile benötigt, um die neuen Kontrollbytes lesen zu können.

Farbenpracht

Sicherlich werden Sie sich bereits gefragt haben, welche Funktion das AT-Bit im zweiten Kontrollwort besitzt. Nun, dieses Bit ist dafür verantwortlich, daß Sprites mit bis zu 16 (!) Farben gleichzeitig dargestellt werden können. Leider muß man hierbei einen weiteren Nachteil in Kauf nehmen, denn um die 16 Farben zu erreichen, müssen immer zwei Sprites miteinander kombiniert werden. Anders ausgedrückt heißt das, daß sich nur noch vier Sprites darstellen lassen. Kombiniert werden können immer nur ein Sprite mit einer geraden und der darauffolgenden ungeraden Nummer, also Sprite 0 mit Sprite 1, Sprite 2 mit Sprite 3 usw. Eine Auswirkung hat das AT-Bit des weiteren auch nur bei den Sprites mit einer ungeraden Nummer (1,3,5,7). Ist es beim Sprite 1 gesetzt, so werden dessen Spritedaten mit denen von Sprite 0 zu einem 4-Bit-Wert verbunden, der 15 verschiedene Kombinationen (Farben) erlaubt. Die Reihenfolge der Bits ist in Tabelle 5 zu sehen. Im übrigen ist es unbedingt erforderlich, daß die beiden Sprites positionsmäßig 100%ig übereinstimmen, da andernfalls das AT-Bit ignoriert wird und die beiden Sprites getrennt behandelt werden.

Aktivierung

Hat man also seine Spritedatenlisten entworfen, wünscht man sich mit Sicherheit nichts sehnlicher herbei, als seine Werke auch auf dem Bildschirm darstellen zu können. Bevor dies aber möglich wird, sind noch ein paar Dinge zu erledigen. Als erstes muß man innerhalb eines Interrupts der dritten Ebene (z.B. dem VBI) oder des Coppers mitteilen, wo die Spritetabellen überhaupt im Speicher zu finden sind. Nebenbei bemerkt: diese Tabellen müssen unbedingt im Chip-Memory liegen, wie das auch bei anderen grafischen und musikalischen Elementen der Fall sein muß. Die Initialisierung von Sprites muß innerhalb einer der aufgezählten Interrupts stattfinden, da der DMA mit jeder abgearbeiteten Spritezeile die Tabellenadressen um ein Wort erhöht. Würde man also am Bildschirmende die Spriteadressen nicht neu beschreiben, ist so gut wie sicher, daß ein heilloses Durcheinander die Folge sein wird, da die Spriteadressen nicht mehr auf die Ursprungsadresse zurückverfrachtet werden. Für jedes der acht darstellbaren Sprites existiert also ein sogenanntes SPRxPT-Register (SpritexPointer), das die Anfangsadresse für das bestimmte Sprite enthält (siehe Tabelle 6). Leider gibt es keine Möglichkeit, die Sprites einzeln zu betätigen, denn alle Sprites werden gleichzeitig über ihren Hauptschalter, der in Bit fünf des DMACON-Registers zu finden ist, entweder ein- oder ausgeschaltet. Um aber kein Durcheinander zu erzeugen, ist es meist notwendig, alle (!) Spriteadressen innerhalb der vertikalen Austastlücke (VBI oder Copper) zu initialisieren, da alle Spriteadressen pro abgearbeitete Zeile immer um ein Wort erhöht werden. Sprites, die nicht auf dem Bildschirm erscheinen sollen, müssen in ihrer Spritedatenliste vier Nullen als Endmarkierung enthalten. Man braucht zwar nicht darzustellende Sprites nicht unbedingt zu initialisieren, jedoch gilt dies immer als die sicherste Methode.
Sind also Copper oder VBI initialisiert, muß nur noch der Sprite-DMA im DMACON-Register gesetzt werden. Grundbedingung, daß aber überhaupt etwas erscheint, ist halt, daß ein eigener Bildschirm definiert wurde.

Bewegung ins Spiel

Bei der Spritebewegung ist zu beachten, daß sie innerhalb eines VBIs erfolgt bzw. daß innerhalb dieses Interrupts die Kontrollwörter der Sprites verändert werden. Ansonsten kann der DMA bei der Darstellung etwas in Unordnung kommen.

Erstes Kontrollwort:

Bit-Nr.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bedeutung	A7	A6	A5	A4	A3	A2	A1	A0	H8	H7	H6	H5	H4	H3	H2	H1

Zweites Kontrollwort:

Bit-Nr.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bedeutung	L7	L6	L5	L4	L3	L2	L1	L0	AT	–	–	–	–	E8	L8	H0

Legende: A0-A8 = Anfangszeile des Sprites (HANF)
H0-H8 = Horizontalposition des Sprites (VANF)
L0-L8 = Endzeile des Sprites + 1 (VEND)
AT = Attach-Kontroll-Bit

Tabelle 3

Beispiel einer Datenliste:

```
Sprite0  dc.w $b096,$b400 ; HANF = $12c, VANF = $b0, VEND = $b4
         dc.w %0000111111110000,%0000000000000000
         dc.w %0001100000011000,%0000011111100000
         dc.w %0001111111111000,%0000011111100000
         dc.w $0000111111110000,%0000000000000000
         dc.w $0,$0; Endmarkierung
```

Tabelle 4

Sprite 0 , erstes Datenwort = Bit 0
" , zweites Datenwort = Bit 1
Sprite 1 , erstes Datenwort = Bit 2
" , zweites Datenwort = Bit 3
Der Wert, der bei der Bitverknüpfung entsteht, zeigt auf die folgende Farbtabelle:

DW3	DW2	DW1	DW0	Register
0	0	0	0	transparent
0	0	0	1	$dff1a2
0	0	1	0	$dff1a4
0	0	1	1	$dff1a6
0	1	0	0	$dff1a8
0	1	0	1	$dff1aa
0	1	1	0	$dff1ac
0	1	1	1	$dff1ae
1	0	0	0	$dff1b0
1	0	0	1	$dff1b2
1	0	1	0	$dff1b4
1	0	1	1	$dff1b6
1	1	0	0	$dff1b8
1	1	0	1	$dff1ba
1	1	1	0	$dff1bc
1	1	1	1	$dff1be

Tabelle 5

Register	Name	Funktion	
$dff120	SPR0PTH	Bits 16-31	der Spritedatenliste
$dff122	SPR0PTL	Bits 0-15	für Sprite-DMA-Kanal 0
$dff124	SPR1PTH	Bits 16-31	der Spritedatenliste
$dff126	SPR1PTL	Bits 0-15	für Sprite-DMA-Kanal 1
$dff128	SPR2PTH	Bits 16-31	der Spritedatenliste
$dff12a	SPR2PTL	Bits 0-15	für Sprite-DMA-Kanal 2
$dff12c	SPR3PTH	Bits 16-31	der Spritedatenliste
$dff12e	SPR3PTL	Bits 0-15	für Sprite-DMA-Kanal 3
$dff130	SPR4PTH	Bits 16-31	der Spritedatenliste
$dff132	SPR4PTL	Bits 0-15	für Sprite-DMA-Kanal 4
$dff134	SPR5PTH	Bits 16-31	der Spritedatenliste
$dff136	SPR5PTL	Bits 0-15	für Sprite-DMA-Kanal 5
$dff138	SPR6PTH	Bits 16-31	der Spritedatenliste
$dff13a	SPR6PTL	Bits 0-15	für Sprite-DMA-Kanal 6
$dff13c	SPR7PTH	Bits 16-31	der Spritedatenliste
$dff13e	SPR7PTL	Bits 0-15	für Sprite-DMA-Kanal 7

Tabelle 6

Wert	Reihenfolge von vorne nach hinten				
0	PLF	SPR0/1	SPR2/3	SPR4/5	SPR6/7
1	SPR0/1	PLF	SPR2/3	SPR4/5	SPR6/7
2	SPR0/1	SPR2/3	PLF	SPR4/5	SPR6/7
3	SPR0/1	SPR2/3	SPR4/5	PLF	SPR6/7
4	SPR0/1	SPR2/3	SPR4/5	SPR6/7	PLF

BPLCON2 $dff104 (Schreibregister)

Bit-Nr.:	15-7	6	5	4	3	2	1	0
Funktion:	Unbel.	PF2PRI	PF2P2	PF2P1	PF2P0	PF1P2	PF1P1	PF1P0

PF2PRI	: Ist dieses Bit im Register gesetzt, erscheint Playfield 2 vor Playfield 1 (siehe unten).
PF1P0 - PF1P2	: Diese drei Bits bilden zusammengefaßt eine 3-Bit-Zahl, die die Position des ersten Playfields zwischen den Spritepaaren enthält. Die gültigen Werte stehen in der obigen Tabelle.
PF2P0 - PF2P2	: Diese Bits haben dieselbe Bedeutung wie bei PF1P0 bis PF1P2, nur daß sie für das zweite Playfield gelten.

Tabelle 7

Wie bei anderen Computern, die in der Lage sind, Sprites darzustellen, existiert auch beim Amiga ein Register, mit dem man den Sprites Prioritäten zuteilen kann.

Prioritäten der Sprites

Genauer gesagt: man kann mit diesem Register bestimmen, an welcher Stelle die Playfields zwischen vier Spritepaaren erscheinen. Die Priorität bei den Sprites selber ist immer dieselbe, das heißt, ein Sprite mit einer niedrigen Nummer erscheint immer vor einem Sprite mit einer noch niedrigeren Nummer, also beispielsweise Sprite 0 vor Sprite 1.

Das Register, das für die Priorität verantwortlich ist, nennt sich BPLCON2 ($DFF104). Es stellt fünf verschiedene Positionen für die Playfields bereit (siehe Tabelle 7).

Sollte der Dual-Playfield-Modus nicht aktiviert sein, haben das PF2PRI-Bit und die Bits PF2P0 bis PF2P2 keine Wirkung, da in diesem Fall die geraden und ungeraden Bitplanes unter PF1P0 bis PF1P2 gleich behandelt werden. Bitte beachten Sie, daß im Dual-Playfield-Modus die Prioritätenanordnung von Playfield 1 stets Vorrang vor der von Playfield 2 besitzt, wenn diese grafische Übereinstimmungen besitzen.

Kollisionen – programmierbarer Konflikt

Kommen wir nun zu einem Abschnitt, der für die programmierenden Spielefreaks sehr interessant ist, sofern sie in ihren Spielen von Sprites und nicht von BOBs Gebrauch machen wollen – Kollisionen.

Das Kollisionsregister wird dann aktiv, wenn sich sichtbare Punkte (die transparenten also ausgenommen) von Sprites oder Playfields treffen. Wie des öfteren werden auch bei diesem Register die Sprites wieder zu Spritepaaren zusammengefaßt (siehe Tabelle 8).

Zu diesem Register muß noch bemerkt werden, daß mit Hilfe eines weiteren Registers erst die Kollision für die Sprites 1, 3, 5, 7 erlaubt werden muß. Dieses gerade eben erwähnte Register hat den Namen CLXCON und ist ein reines Schreibregister, das heißt, es können keine Werte aus diesem Wortregister gelesen werden (Tabelle 9 zeigt das Schreibregister). Die Bits 0 bis 11 haben eine besondere Bedeutung. Mit ihrer Hilfe läßt sich einstellen, welche Farben für eine Kollision zugelassen sind. Die ENBPx-Bits sind für die Zulassung der MVBPx-Bits zuständig. Bei jedem gesetzten ENBPx-Bit wird auch das entsprechende MVBPx-Bit erlaubt bzw. zugelassen. Ist ein ENBPx-Bit hingegen nicht gesetzt, dient das entsprechende MVBPx-Bit als Joker, das heißt, es kann entweder gesetzt oder nicht gesetzt sein. Nur als Ergänzung: Die Bit-Muster, die mit den Bits 0 bis 11 erzeugt werden, geben an, welche Farben für eine Kollision herangezogen werden. Zum Abschluß und zum besseren Verständnis deshalb drei Beispiele in Tabelle 10.

OS-Routinen

Das Operating System (Betriebssystem) des Amiga bietet einige Routinen, die für die Benutzung der Sprites

Kollisionsregister CLXDAT ($DFF00E) (Leseregister)

Bit-Nr.	Kollision zwischen
15	Unbenutzt
14	Sprite 4/5 und Sprite 6/7
13	Sprite 2/3 und Sprite 6/7
12	Sprite 2/3 und Sprite 4/5
11	Sprite 0/1 und Sprite 6/7
10	Sprite 0/1 und Sprite 4/5
9	Sprite 0/1 und Sprite 2/3
8	Playfield 2 (gerade Bitplanes) und Sprite 6/7
7	Playfield 2 (gerade Bitplanes) und Sprite 4/5
6	Playfield 2 (gerade Bitplanes) und Sprite 2/3
5	Playfield 2 (gerade Bitplanes) und Sprite 0/1
4	Playfield 1 (ungerade Bitplanes) und Sprite 6/7
3	Playfield 1 (ungerade Bitplanes) und Sprite 4/5
2	Playfield 1 (ungerade Bitplanes) und Sprite 2/3
1	Playfield 1 (ungerade Bitplanes) und Sprite 0/1
0	Playfield 1 und Playfield 2

Tabelle 8

CLXCON ($DFF098) Schreibregister:

Bit-Nr.	Name	Funktion
15	ENSP7	Kollisionserkennung für Sprite 7 erlauben
14	ENSP5	Kollisionserkennung für Sprite 5 erlauben
13	ENSP3	Kollisionserkennung für Sprite 3 erlauben
12	ENSP1	Kollisionserkennung für Sprite 1 erlauben
11	ENBP6	Bitplane 6 mit MVBP6 vergleichen
10	ENBP5	Bitplane 5 mit MVBP5 vergleichen
9	ENBP4	Bitplane 4 mit MVBP4 vergleichen
8	ENBP3	Bitplane 3 mit MVBP3 vergleichen
7	ENBP2	Bitplane 2 mit MVBP2 vergleichen
6	ENBP1	Bitplane 1 mit MVBP1 vergleichen
5	MVBP6	Wert für die Kollision mit Bitplane 6
4	MVBP5	Wert für die Kollision mit Bitplane 5
3	MVBP4	Wert für die Kollision mit Bitplane 4
2	MVBP3	Wert für die Kollision mit Bitplane 3
1	MVBP2	Wert für die Kollision mit Bitplane 2
0	MVBP1	Wert für die Kollision mit Bitplane 1

Tabelle 9

programmiert wurden. Allerdings sind diese so unkomfortabel, daß sie so gut wie gar nicht benutzt werden; in den neuesten Games sowieso nicht (häufige Ausnahme: C-Programme). Aus diesem Grund ist es das beste, selbst die Dinge in die Hand zu nehmen. Wie das geht, haben Sie ja in diesem Artikel kennengelernt.

Beispiel

Nichts macht ein Problem so verständlich wie ein Beispiel. Aus diesem Grunde haben wir auf den nächsten Seiten ein Beispiellisting abgedruckt. Um es aber nicht noch länger zu machen, als es ohnehin schon ist (wenn man nämlich an die langen Abtippzeiten denkt), habe ich auf die Vorführung eines Multicolorsprites verzichtet und lieber ein normales Sprite genommen.

Damit Sie schon im voraus wissen, was Sie erwartet, sei gesagt, daß sich ein rotierender Pfeil nach Start des Programms im Kreis um den Bildmittelpunkt dreht. Ein Interrupt der dritten Ebene sorgt für die Bewegung und die Animation.

Damit möchte ich diesen Spritekurs abschließen und hoffe, daß Ihnen meine Erklärungen geholfen haben.

(mm)

ENBPx	MVBPx	Kollision bei Bitmuster
001111	xx1001	001001,011001,101001,111001
000000	xxxxxx	jedes Bitmuster möglich
110111	11x000	110000,111000

Tabelle 10

Listings

```
             opt   l-

;
; SPRITE-Kurs Demolisting fuer DEVPAC II
; (c) 1989 by Ulf Petersen

COP1LC                EQU     $80
COP2LC                EQU     $84
COPJMP1               EQU     $88
COPJMP2               EQU     $8A

BPLCON0               EQU     $100
BPLCON1               EQU     $102
BPLCON2               EQU     $104
BPL1PTH               EQU     $E0
BPL1PTL               EQU     $E2
BPL1MOD               EQU     $108
BPL2MOD               EQU     $10A
DIWSTRT               EQU     $8E
DIWSTOP               EQU     $90
DDFSTRT               EQU     $92
DDFSTOP               EQU     $94
SPR0PTH               EQU     $120
SPR0PTL               EQU     $122
COLOR17               EQU     $1A2

DMACON                EQU     $96
CIA                   EQU     $BFE001

FORBID                EQU     -30-102
PERMIT                EQU     -30-108
ALLOCMEM              EQU     -30-168
FREEMEM               EQU     -30-180
OPENLIB               EQU     -30-522
EXEC                  EQU     4
CHIP                  EQU     2
CLEAR                 EQU     CHIP+$10000
STLIST                EQU     38

Plane                 EQU     256*40
Copsize               EQU     96
SpLength              EQU     52

; Speicherbereiche fuer Copperliste, Spritedatenliste
; und Bitplane reservieren lassen

start                 move.l  EXEC,a6
                      move.l  #Copsize,d0      ;Speicher fue
  r Copperliste
                      move.l  #CLEAR,d1
                      jsr     ALLOCMEM(a6)
```

Listing Sprite

```
                      move.l  d0,Copadr

                      move.l  #SpLength,d0     ;Speicher fue
  r Spriteliste
                      move.l  #CLEAR,d1
                      jsr     ALLOCMEM(a6)
                      move.l  d0,Spadr
                                               ;Adresse der
  Spriteliste
                      move.w  d0,COPLIST+14    ;in die Copp
  erliste
                      swap    d0               ;eintragen
                      move.w  d0,COPLIST+10

                      move.l  #Plane,d0        ;Speicher fue
  r Bitplane
                      move.l  #CLEAR,d1
                      jsr     ALLOCMEM(a6)
                      move.l  d0,Planeadr
                                               ;Bitplaneadre
  sse in die
                      move.w  d0,COPLIST+6     ;Copperliste
  eintragen
                      swap    d0
                      move.w  d0,COPLIST+2

                      move.l  #Copsize-1,d0    ;Copperliste
  ins CHIP-Ram
                      lea     COPLIST,a0       ;kopieren
                      move.l  Copadr,a1
Copy1                 move.b  (a0)+,(a1)+
                      dbf     d0,Copy1

                      move.l  #SpLength-1,d0 ;Spritedatenl
  iste ins
                      lea     SPRITETAB,a0     ;CHIP-Ram kop
  ieren
                      move.l  Spadr,a1
Copy2                 move.b  (a0)+,(a1)+
                      dbf     d0,Copy2

                      jsr     FORBID(a6)       ;TASK-Switchi
  ng
                      lea     $dff000,a5       ;sperren und
  neue
                      move.w  #$1e0,DMACON(a5) ;COPPER-Lis
  te installieren
                      move.l  Copadr,COP1LC(a5)
                      clr.w   COPJMP1(a5)

move.w                #$2981,DIWSTRT(a5)       ;Bitplanepara
  meter
                      move.w  #$29c1,DIWSTOP(a5) ;einstell
  en
```

Listing Sprite

```
                move.w  #$0038,DDFSTRT(a5)
                move.w  #$00d0,DDFSTOP(a5)
                move.w  #%0001001000000000,BPLCON0(a5)
                clr.w   BPLCON1(a5)
                clr.w   BPLCON2(a5)
                clr.w   BPL1MOD(a5)
                clr.w   BPL2MOD(a5)
                move.w  #%1000001110100000,DMACON(a5) ;Bildschirm, Copper und
                                        ;Sprites einschalten
                move.l  $6C,OldInt      ;Interrupt-Vektor
                move.l  #NewInt,$6c     ;verbiegen
wait            btst    #6,CIA          ;Linke Maustaste
                bne.s   wait            ;gedrueckt ?
                move.l  OldInt,$6C      ;Alter IRQ-Vektor
                move.l  #GrafLib,a1
                clr.l   d0
                jsr     OPENLIB(a6)     ;GFX-Library oeffnen
                move.l  d0,a4
                move.l  STLIST(a4),COP1LC(a5) ;Alte COPPER-Liste
                clr.w   COPJMP1(a5)
                move.w  #$83e0,DMACON(a5)
                jsr     PERMIT(a6)      ;TASK-Switching erlauben
;Abschliessend die Anfangs belegten
;Speicherbereiche wieder freigeben
                move.l  Copadr,a1
                move.l  #Copsize,d0
                jsr     FREEMEM(a6)
                move.l  Spadr,a1
                move.l  #SpLength,d0
                jsr     FREEMEM(a6)
                move.l  Planeadr,a1
                move.l  #Plane,d0
                jsr     FREEMEM(a6)
                clr.l   d0              ;Alles klar !
                rts                     ;Ab in den CLI
                                        ;Adressen im CHIP-Ram
Copadr          dc.l    0               ;Copperliste
Planeadr        dc.l    0               ;Bitplane
Spadr           dc.l    0               ;Spriteliste
                                        ;Neuer Interrupt
NewInt          movem.l d0-d7/a0-a6,-(sp)
                add.w   #1,Flag         ;Verzoegerung, damit
                cmp.w   #5,Flag         ;IRQ nur alle 5/50stel s
                bne     Goto3           ;aktiviert wird
                move.w  #0,Flag
                lea     BewTab,a1       ;Aktuelle Animationsfigur
                add.l   Figur,a1        ;in die Spritedaten-
                moveq   #43,d0          ;liste kopieren
Holen           move.l  (a1),a0
                move.l  Spadr,a1
                add.l   #4,a1
Copy3           move.b  (a0)+,(a1)+
                dbf     d0,Copy3
                add.l   #4,Figur
                cmp.l   #104,Figur      ;Wenn alle Figuren gezeigt,
                bne     Goto1           ;dann wieder von vorne
                move.l  #0,Figur
Goto1           move.l  Position,a1     ;Neue Position des Sprites
                move.b  (a1),d0         ;mit Hilfe der Tabelle
                add.b   #$82,d0         ;errechnen und in die
                move.l  Spadr,a0        ;Spritedatenliste eintragen
                move.b  d0,1(a0)
                add.l   #1,Position
                cmp.l   #BewTab,Position ;Tabellenende ?
                bne     Not
                move.l  #POSTAB,Position
Not             cmp.l   #28,Figur       ;Wenn 7te Figur
                bne     Goto2           ;erreicht, dann
                move.l  Copadr,a0       ;rote Farbtoene fuers
                move.w  #$0f00,74(a0)   ;Sprite in die COPPER-
                move.w  #$0b00,78(a0)   ;liste eintragen
                move.w  #$0700,82(a0)
Goto2           cmp.l   #84,Figur       ;s.o., nur mit Blau
                bne     Goto3           ;und bei 21ter Figur
                move.l  Copadr,a0
                move.w  #$000f,74(a0)
                move.w  #$000b,78(a0)
                move.w  #$0007,82(a0)
Goto3           movem.l (sp)+,d0-d7/a0-a6
                dc.w    $4ef9           ;Code fuerJMP
OldInt          dc.l    0               ;Alter Vektor
Flag            dc.w    0               ;Time-Counter fuer IRQ
                                        ;Spritedatenliste
SPRITETAB       dc.w    $a000,$ab00     ;VANF=$a0 ,VEND=$ab
                dc.l    0,0,0,0,0,0,0,0,0,0
                dc.w    0,0             ;Endmarkierung
;Vorgefertigte COPPER-Liste. Muss noch install.werden.
COPLIST         dc.w    BPL1PTH,$0000,BPL1PTL,$0000
                dc.w    SPR0PTH,$0000,SPR0PTL,$0000
                dc.w    SPR0PTH+4,$0000,SPR0PTL+4,$0000
                dc.w    SPR0PTH+8,$0000,SPR0PTL+8,$0000
                dc.w    SPR0PTH+12,$0000,SPR0PTL+12,$0000
                dc.w    SPR0PTH+16,$0000,SPR0PTL+16,$0000
                dc.w    SPR0PTH+20,$0000,SPR0PTL+20,$0000
                dc.w    SPR0PTH+24,$0000,SPR0PTL+24,$0000
                dc.w    SPR0PTH+28,$0000,SPR0PTL+28,$0000
                dc.w    COLOR17,$000f,COLOR17+2,$000b
                dc.w    COLOR17+4,$0007
                dc.w    $FFFF,$FFFE
GrafLib         dc.b    "graphics.library",0
                cnop    0,2
Position        dc.l    POSTAB
;Positionstabelle fnr das Sprite
POSTAB          dc.b    14,13,12,11,10,9,8,7,6,5,4,3,2,1
                dc.b    2,3,4,5,6,7,8,9,10,11,12,13
;Tabelle, die angibt in welcher Reihenfolge die Animations-
;figuren angezeigt werden
BewTab          dc.l    Phase1,Phase2,Phase3,Phase4
                dc.l    Phase5,Phase6,Phase7,Phase14
                dc.l    Phase13,Phase12,Phase11,Phase10
                dc.l    Phase9,Phase8,Phase9,Phase10
                dc.l    Phase11,Phase12,Phase13,Phase14
                dc.l    Phase7,Phase6,Phase5,Phase4
                dc.l    Phase3,Phase2
Figur           dc.l    0               ;Aktuelle Animationsfigur
;Es folgen die verschiedenen Bewegungsphasen
;zur Verdeutlichung als Mini-Bitplanes
Phase1          dc.w    %0000100000000000,%0000000000000000
                dc.w    %0001100000000000,%0000010000000000
                dc.w    %0011111111111100,%0000000000000000
                dc.w    %0111111111111100,%0000000000000010
                dc.w    %1111111111111101,%0000000000000011
                dc.w    %0111111111111101,%0000000000000011
                dc.w    %0011111111111101,%0000000000000011
                dc.w    %0001100000000001,%0000011111111111
                dc.w    %0000101111111111,%0000011111111111
                dc.w    %0000001000000000,%0000011000000000
                dc.w    %0000001000000000,%0000001000000000
Phase2          dc.w    %0000100000000000,%0000000000000000
                dc.w    %0001100000000000,%0000010000000000
                dc.w    %0001111111111000,%0000000000000000
                dc.w    %0011111111111000,%0000000000000100
                dc.w    %0111111111111010,%0000000000000110
                dc.w    %0011111111111010,%0000000000000110
                dc.w    %0001111111111010,%0000000000000110
                dc.w    %0001100000000010,%0000011111111110
                dc.w    %0000101111111110,%0000011111111110
                dc.w    %0000001000000000,%0000011000000000
                dc.w    %0000001000000000,%0000001000000000
```

Listing Sprites

```
Phase3          dc.w    %0000010000000000,%0000000000000000
                dc.w    %0000110000000000,%0000001000000000
                dc.w    %0000111111110000,%0000000000000000
                dc.w    %0001111111110000,%0000000000001000
                dc.w    %0011111111110100,%0000000000001100
                dc.w    %0001111111110100,%0000000000001100
                dc.w    %0000111111110100,%0000000000001100
                dc.w    %0000110000000100,%0000001111111100
                dc.w    %0000010111111100,%0000001111111100
                dc.w    %0000000100000000,%0000001100000000
                dc.w    %0000000100000000,%0000000100000000

Phase4          dc.w    %0000010000000000,%0000000000000000
                dc.w    %0000110000000000,%0000001000000000
                dc.w    %0000111111100000,%0000000000000000
                dc.w    %0001111111100000,%0000000000010000
                dc.w    %0001111111101000,%0000000000011000
                dc.w    %0001111111101000,%0000000000011000
                dc.w    %0000111111101000,%0000000000011000
                dc.w    %0000110000001000,%0000001111111000
                dc.w    %0000010111111000,%0000001111111000
                dc.w    %0000000100000000,%0000001100000000
                dc.w    %0000000100000000,%0000000100000000

Phase5          dc.w    %0000010000000000,%0000000000000000
                dc.w    %0000110000000000,%0000001000000000
                dc.w    %0000111111000000,%0000000000000000
                dc.w    %0000111111000000,%0000000000100000
                dc.w    %0000111111010000,%0000000000110000
                dc.w    %0000111111010000,%0000000000110000
                dc.w    %0000111111010000,%0000000000110000
                dc.w    %0000110000010000,%0000001111110000
                dc.w    %0000010111110000,%0000001111110000
                dc.w    %0000000100000000,%0000001100000000
                dc.w    %0000000100000000,%0000000100000000

Phase6          dc.w    %0000010000000000,%0000000000000000
                dc.w    %0000010000000000,%0000001000000000
                dc.w    %0000011110000000,%0000000000000000
                dc.w    %0000011110000000,%0000000001000000
                dc.w    %0000011110100000,%0000000001100000
                dc.w    %0000011110100000,%0000000001100000
                dc.w    %0000011110100000,%0000000001100000
                dc.w    %0000010000100000,%0000001111100000
                dc.w    %0000010111100000,%0000001111100000
                dc.w    %0000000100000000,%0000001100000000
                dc.w    %0000000100000000,%0000000100000000

Phase7          dc.w    %0000001000000000,%0000000000000000
                dc.w    %0000001000000000,%0000000100000000
                dc.w    %0000001100000000,%0000000000000000
                dc.w    %0000001100000000,%0000000010000000
                dc.w    %0000001101000000,%0000000011000000
                dc.w    %0000001101000000,%0000000011000000
                dc.w    %0000001101000000,%0000000011000000
                dc.w    %0000001001000000,%0000000111000000
                dc.w    %0000001011000000,%0000000111000000
                dc.w    %0000000010000000,%0000000110000000
                dc.w    %0000000010000000,%0000000010000000

;Die restlichen Phasen in HEX-Form, da sie nur umgedreht wurden

Phase8          dc.l    $00400000,$00600000,$fff00000,$fff80000
                dc.l    $fffc0000,$fff80006,$fff3000f,$00667f9e
                dc.l    $3fcc3fbc,$00180038,$00100010

Phase9          dc.l    $00400000,$00600000,$7fe00010,$7ff80008
                dc.l    $7ff80000,$7ff0000c,$7fe6001e,$006c3f9c
                dc.l    $1fc81fb8,$00180038,$00100010

Phase10         dc.l    $00800000,$00c00000,$3fc00020,$3ff00010
                dc.l    $3ff00000,$3fe00018,$3fcc003c,$00d81f38
                dc.l    $0f900f70,$00300070,$00200020

Phase11         dc.l    $00800000,$00c00000,$1fc00020,$1ff00010
                dc.l    $1fe00010,$1fe80018,$1fc80038,$00d80f38
                dc.l    $07900770,$00300070,$00200020

Phase12         dc.l    $00800000,$00c00000,$0fc00020,$0fd00030
                dc.l    $0fd00030,$0fd00030,$0fd00030,$00d00730
                dc.l    $03900370,$00300070,$00200020

Phase13         dc.l    $00800000,$00800040,$07a00060,$07a00060
                dc.l    $07a00060,$07a00060,$07a00060,$00a00360
                dc.l    $01a00160,$00200060,$00200020

Phase14         dc.l    $01000000,$01000080,$034000c0,$034000c0
                dc.l    $034000c0,$034000c0,$034000c0,$014000c0
                dc.l    $014000c0,$004000c0,$00400040
```

Listing Sprites

Garry Glendown

SprEd – das Sprite-Editor-Projekt

Teil 2: Malfunktionen

Wir können zwar schon Sprites Punkt für Punkt malen, abspeichern und wieder einladen, doch ist das natürlich nicht der Weisheit letzter Schluß. Daher widmen wir uns heute einigen Malfunktionen sowie dem Abspeichern eines Sprites als Source.

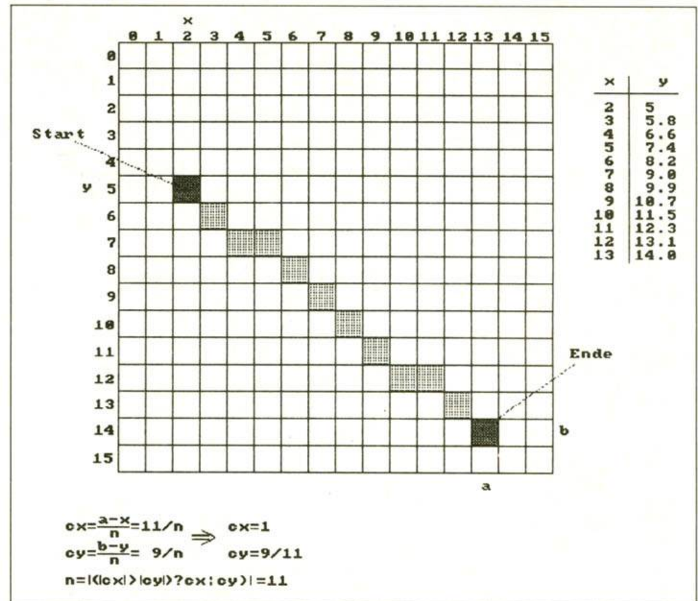


Bild 1: Zeichnen einer Linie mit der Gummibandmethode

Um die Bewegungen der Maus korrekt zu behandeln, ist eine der wichtigsten Routinen in diesem Teil die Funktion 'HandleMovement'. Sie wird aufgerufen, wenn vom Hauptprogramm eine MOUSEMOVE-Message erhalten wurde.

Sie ermöglicht auch ein 'Rubber Banding', also eine Art Gummiband, das anstelle der kompletten Linie gezeichnet wird. In der Routine wird geprüft, ob die nächste IntuiMessage ebenfalls vom Typ 'MOUSEMOVE' ist. Ist dies der Fall, so wird nur eine dünne Linie (bzw. ein dünnes Rechteck) gezeichnet. Andernfalls muß die Linie oder das Rechteck 'normal' gezeichnet werden, da die Maus nicht bewegt wurde.

Doch nun zu den eigentlichen Malroutinen. Als erstes wäre da die Routine zum Malen eines Rechtecks. Das Rechteck besteht aus zwei Paaren von parallelen Linien, so daß man durch Verwendung von zwei Schleifen, die jeweils in gleicher Höhe bzw. Breite waagrechte und senkrechte Linien ziehen, leicht ein Rechteck erhält. Schwieriger wird es jedoch dann, wenn man bei der Line-Funktion auch schiefe Linien erhalten möchte.

Zum Zeichnen einer Linie wird innerhalb von SprEd auf 'Float'-Variablen zurückgegriffen, da Schrittweiten zwischen 0 und 1 benötigt werden. Außerdem werden temporäre Variablen 'cx' und 'cy' verwendet, die die Koordinaten des aktuell zu setzenden Punktes enthalten. Zu diesen werden dann jeweils auch die Schritte in X- und Y-Richtung hinzugezählt. Zur Veranschaulichung möchten wir auf die Grafik in Bild 1 verweisen.

Die Hauptschleife wird beim Zeichnen so oft durchlaufen, wie die größte Entfernung zwischen den Punkten in X- bzw. Y-Richtung ist, in unserem Beispiel beträgt sie '11'; die Schleife läuft also von 0 bis 11 inklusive, also 12 Durchläufe. Innerhalb der Schleife wird zunächst der Punkt (rn(cx),rn(cy)) gesetzt. Danach werden zu den beiden Koordinaten 'xc' und 'yc' hinzuaddiert. Durch das Runden mittels der Funktion 'rn' wird zum einen die Float-Variable in eine Integer-Variable umgewandelt, zum anderen wird zu dem Ausgangswert '0.5' hinzuaddiert, wodurch jeweils der naheliegendste Punkt gesetzt wird. Zu beachten ist außerdem noch die Variable 'final'. Enthält sie einen Wert ungleich '0', so werden die Punkte nicht nur am Bildschirm, sondern auch im Datenfeld gesetzt.

Des weiteren sorgt die Variable 'c' dafür, daß die Punkte auf einer Linie auch wieder auf ihren Ursprungswert gesetzt werden können, womit die Linie wieder gelöscht wird.

Hilfe, es läuft über...

Doch nun zu dem interessantesten Teil, der Füllroutine. Während der Entstehungsphase von SprEd standen zwei Möglichkeiten zur Auswahl: entweder man benutzt die 'Flood'-Routine vom Betriebssystem, oder aber man schreibt eine eigene Routine, die diese Arbeit erledigt.

Die Benutzung der Systemroutine schien in diesem Fall nicht unbedingt sinnvoll, da das zu füllende Feld recht klein ist, und damit der zusätzliche Aufwand den Gewinn an Zeit aufwiegen würde. Somit nehmen wir an dieser Stelle den zweiten Weg.

Beim Füllen wird wie folgt vorgegangen. Ein Punkt auf der Malfläche besitzt zwei Zustände: Entweder er hat eine Farbe, oder er ist mit Farbe '128' markiert. Nachdem der

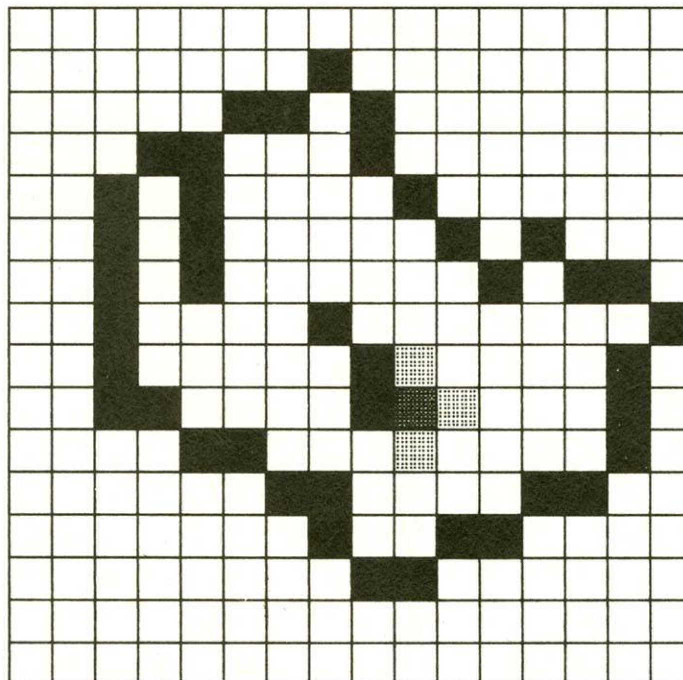


Bild 2: Veranschaulichung für das Ausfüllen einer Fläche

Wichtig!

Das Programm SprEd benötigt auf jeden Fall die Arp-Library. Diese ist Public Domain und darf deswegen nicht abgedruckt werden. Wenn Sie SprEd lauffähig machen wollen, besorgen Sie sich bitte zuerst die Arp-Library. Sie befindet sich unter anderem auf der Fishdisk 123 und auf der Panorama27c.

```
# Alle Includes wurden
mit der Option
' + hlib:all.pre'
vorcompiliert

.c.o:
cc $*.c + ilib:all.pre
spred: spred.o part2.o
ln + q + cd $*.o part2.o -
la -lm -lc
```

Ausgangspunkt, der mit der Maus gewählt wurde, auf '128' gesetzt ist, beginnt eine Schleife, die jeweils alle 256 Punkte durchläuft. Befindet sich in wenigstens einem Punkt dieses Feldes einer mit der Markierung '128', so werden senkrecht und waagrecht von diesem Punkt aus alle vier Punkte ebenfalls mit der Farbe '128' markiert, sofern diese die zu füllende Farbe besitzen. Diese Schleife wird solange wiederholt, bis bei einem Durchlauf kein '128'er Feld gefunden wird. Um auch dem Auge etwas zu bieten, wird die gefüllte Fläche nicht erst am Ende gezeichnet, sondern man sieht auch während des Füllens den Fortschritt. Dies hat sich angeboten, da bei dem kleinen Feld kein großer Zeitverlust zu erwarten war.
Zu beachten ist übrigens noch, daß rund um die temporäre Arbeitsfläche ein Ring von '127'er Feldern gelegt wurde, um so ein 'Ausbrechen' der Füllroutine aus dem Füllfeld zu verhindern.
Um sich die Arbeitsweise noch weiter zu veranschaulichen, ist in Bild zwei eine Fläche abgebildet, die man anhand der Beschreibung einmal per Hand ausfüllen kann...

Wo bleibt mein Source?

Als Abschluß der heutigen Folge kommt nun noch der Source-Generator dran. Durch Drücken des Gadgets 'Make Src' wird ein Filerequester geöffnet, bei dem man den Namen des Source-Files eingeben kann. Danach wird das aktuelle Sprite als C-Source im File abgespeichert. Vor der Einbindung in ein C-Programm braucht man nun nur noch den Namen des Feldes zu ändern (sofern mehrere Sprites verwendet werden), und schon kann man loslegen.

(br)



Bitte die Zeilennummern nicht mit abtippen, sie dienen lediglich der Übersicht

Änderungen der Files von SprEd in Ausgabe 2/90:

defines.h:
neue Zeile:
```
#define SRCFLG 22
```

spred.c:
Zeile 72:
OpenStuff();
ersetzen durch:
```
Mode = DRAW;
OpenStuff();
DisplayMode("Draw ");
```
Zeile 86:
nach DRAWIT = FALSE;
einsetzen:
```
switch (Mode) {
case LINE:
FinishLine();
break;
case RECT:
FinishRect();
break;
}
LINEDRAWN = FALSE;
```
Zeile 97:
nach
DRAWIT = FALSE;
einsetzen:
```
switch (Mode) {
case LINE:
FinishLine();
break;
case RECT:
FinishRect();
break;
}
LINEDRAWN = FALSE;
```
Zeile 236:
Palette();
ersetzen durch:
```
DoPalette();
```
Zeilen 254 - 262 ersetzen durch:
```
RefreshColor();
```
Zeile 290:
if (DRAWIT) {
ersetzen durch:
```
if
(DRAWIT&&(Mode = = DRAW-
)) {
```
nach Zeile 304 ergänzen:
```
case 'L':
DoLine();break;
case 'D':
DoDraw();
break;
case 'F':
DoFill();
break; case 'C':
DoCirc();
break;
case 'R':
DoRect();
break;
case '>':
CurCol + + ;
if (CurCol > 15) CurCol = 0;
RefreshColor();
break;
case '<':
CurCol - ;
if (CurCol < 0) CurCol = 15;
RefreshColor();
break;
```
Zeilen 402 - 418 entfallen



Listings 

```
#include "defines.h"


BOOL LINEDRAWN,BLINEDRAWN,DRAWOK=FALSE;
extern struct Window *win;

extern UBYTE image[20][16][16],undo[16][16];
extern int CurImage,CurCol;
extern BOOL DRAWIT,ERASE;
extern int Mode;
extern struct FileHandle *fh,*Open();
extern char *DoFile();

extern struct RastPort *rp;

int p1x,p1y,p2x,p2y; /* Punkte für Linien, Box, Circle */


/****************************************************\
 *          Abfrage der Maus-Koordinaten             *
\****************************************************/

Pressed(x,y)
SHORT x,y;
{
 if ((x>=55)&&(x<375)) {
  if ((y>=18)&&(y<178)) {
   switch (Mode) {

    case DRAW:
     if (ERASE)
      PlotIt(x,y,0);
     else
      PlotIt(x,y,CurCol);
     break;

    case FILL:
     if (ERASE)
      Fill(x,y,0);
     else
      Fill(x,y,CurCol);
     break;

    case LINE:
    case RECT:
     LINEDRAWN=FALSE;
     DRAWOK=TRUE;
     conv(x,y,&p1x,&p1y);
     p2x=p1x;
     p2y=p1y;
     if (ERASE)
      PlotPoint(p1x,p1y,0);
     else
      PlotPoint(p1x,p1y,CurCol);
     break;
   }
  }
 }
}

DisplayAbout()
{}

ShowError(txt)
char *txt;
{}

/****************************************************\
 *  DoFill / DoCirc / DoRect / DoLine / DoDraw       *
 *                                                   *
 *  Diese Funktionen setzen die 'Mode'-Variable für  *
 *              weitere Verarbeitung                 *
\****************************************************/

DoFill()
{
 Mode=FILL;
 DisplayMode("Fill  ");
}

DoCirc()
{
 Mode=CIRC;
 DisplayMode("Circle");
}

DoRect()
{
 Mode=RECT;
 DisplayMode("Rect  ");
}

DoLine()
{
 Mode=LINE;
 DisplayMode("Line  ");
}

DoDraw()
{
 Mode=DRAW;
 DisplayMode("Draw  ");
}

DoPalette()
{}

DoAnimate()
{}

/****************************************************\
 *                   CreateSrc                       *
 *    Hiermit wird ein File-Requester angezeigt      *
 *        Wird ein Filename eingegeben,              *
 *          so wird ein Sourcefile erzeugt.          *
\****************************************************/

CreateSrc()
{
char *c,fname[256],oline[256];
long l;
int t,g;
USHORT tmp[16];
 fname[0]='\0';
 c=DoFile((UBYTE *)"Sourcefile sichern:",fname);
 if (c) {

  fh=Open(fname,MODE_NEWFILE);
  if (fh) {

   l=Write(fh,"/* Sprite erzeugt mit SprEd (C) Amiga DOS/GG */\n\n",48L);
   l+=Write(fh,"USHORT sprite[]={\n",18L);
   for (t=0;t<4;t++) {

    CalcPlane(1<<t,tmp);

    for (g=0;g<16;g+=4) {
     sprintf(oline,"\t0x%04x,0x%04x,0x%04x,0x%04x",tmp[g],tmp[g+1],tmp[g+2],tmp[g+3]);
     l+=Write(fh,oline,28L);
     if ((g==12)&&(t==3)) l+=Write(fh,"};\n\n",4L);
     else l+=Write(fh,",\n",2L);
    }
   }

   Close(fh);
   if (l!=548L) ShowError("Fehler beim Speichern!");
  }
 }
}

/****************************************************\
 *                   CalcPlane                       *
 *          berechnet die USHORT-Werte               *
 *          für die einzelnen Bitplanes              *
\****************************************************/

CalcPlane(p,f)
int p;
USHORT *f;
{
int t,g;
USHORT v;

 for (t=0;t<16;t++) {
  v=0;

  for (g=0;g<16;g++) {
   if (image[CurImage][g][t]&p) v|=(1<<(15-g));
  }
  f[t]=v;
 }
}

/****************************************************\
 *                HandleMovement                     *
 *                                                   *
 *  sorgt für die Bearbeitung der MOUSEMOVE-Events   *
\****************************************************/

HandleMovement(x,y)
int x,y;
{
struct IntuiMessage *m;
struct List *l;
 if (DRAWOK)
 switch(Mode) {

  case LINE:

   if (LINEDRAWN) {
    DrawLine(p1x,p1y,p2x,p2y);
    LINEDRAWN=FALSE;
   }

   if (BLINEDRAWN) {
    MakeLine(p1x,p1y,p2x,p2y,0,0,0);
```

Listing Part2.c

Listing Part2.c

```
    BLINEDRAWN=FALSE;
   }

   if (DRAWIT) {
    l=&(win->UserPort->mp_MsgList);
    if(l->lh_Head) {
     m=(struct IntuiMessage *)l->lh_Head;

     if (m->Class==MOUSEMOVE) {
      LINEDRAWN=TRUE;
      conv(x,y,&p2x,&p2y);
      DrawLine(p1x,p1y,p2x,p2y);
     }
     else {
      if (ERASE)
       MakeLine(p1x,p1y,p2x,p2y,1,0,0);
      else
       MakeLine(p1x,p1y,p2x,p2y,1,0,CurCol);
     }
    }
    else {
     if (ERASE)
      MakeLine(p1x,p1y,p2x,p2y,1,0,0);
     else
      MakeLine(p1x,p1y,p2x,p2y,1,0,CurCol);
    }
   }
   break;

  case RECT:
   if (LINEDRAWN) {
    DrawRect(p1x,p1y,p2x,p2y);
    LINEDRAWN=FALSE;
   }

   if (BLINEDRAWN) {
    MakeRect(p1x,p1y,p2x,p2y,0,0,0);
    BLINEDRAWN=FALSE;
   }

   if (DRAWIT) {
    l=&(win->UserPort->mp_MsgList);

    if(l->lh_Head) {
     m=(struct IntuiMessage *)l->lh_Head;
     if (m->Class==MOUSEMOVE) {
      LINEDRAWN=TRUE;
      conv(x,y,&p2x,&p2y);
      DrawRect(p1x,p1y,p2x,p2y);
     }
     else {
      if (ERASE)
       MakeRect(p1x,p1y,p2x,p2y,1,0,0);
      else
       MakeRect(p1x,p1y,p2x,p2y,1,0,CurCol);
     }
    }
    else {
     if (ERASE)
      MakeRect(p1x,p1y,p2x,p2y,1,0,0);
     else
      MakeRect(p1x,p1y,p2x,p2y,1,0,CurCol);
    }
   }
   break;
 }
}

/*****************************************************\
 *                    DrawLine                       *
 *                                                   *
 *     Zeichnet das 'Gummiband' für die Linie        *
\*****************************************************/

DrawLine(xf,yf,xt,yt)
int xf,xt,yf,yt;
{
long x,y;
 SetAPen(rp,15L);
 SetDrMd(rp,COMPLEMENT);

 convpos(xf,yf,&x,&y);
 Move(rp,x,y);
 convpos(xt,yt,&x,&y);
 Draw(rp,x,y);

 SetAPen(rp,(long)CurCol);
 SetDrMd(rp,JAM2);
}

/*****************************************************\
 *                    DrawRect                       *
 *                                                   *
 *    Zeichnet das 'Gummiband' für das Rechteck      *
\*****************************************************/

DrawRect(xt,yt,xb,yb)
int xt,yt,xb,yb;
{
long x,y,a,b;
```

Listing Part2.c

```
 SetAPen(rp,15L);
 SetDrMd(rp,COMPLEMENT);

 convpos(xt,yt,&x,&y);
 convpos(xb,yb,&a,&b);

 Move(rp,x,y);
 Draw(rp,a,y);
 Draw(rp,a,b);
 Draw(rp,x,b);
 Draw(rp,x,y);

 SetAPen(rp,(long)CurCol);
 SetDrMd(rp,JAM2);
}

/*****************************************************\
 *                  plus / iplus                     *
 *                                                   *
 *  Berechnet den Betrag eines Float/Integer-Wertes  *
\*****************************************************/

float plus(x)float x;{return((x>=0)?x:-x);}
int iplus(x){return((x>=0)?x:-x);}

/*****************************************************\
 *                       rn                          *
 *                                                   *
 *              rundet ein Float-Zahl                *
\*****************************************************/

int rn(x)float x;{return((int)(x+(float).5));}

/*****************************************************\
 *                    MakeRect                       *
 *                                                   *
 *     erzeugt das fertige Rechteck im Mal-Feld      *
\*****************************************************/

MakeRect(x,y,a,b,c,final,color)
int x,y,a,b,c,final,color;
{
int xc,yc,cx,cy,t,n;
 xc=x;
 yc=y;
 n=((a-x)>0)?1:-1;

 for (t=0;t<=iplus(a-x);t++) {
  if (c) {
   if (final) {
    image[CurImage][xc][y]=color;
    image[CurImage][xc][b]=color;
   }
   PlotPoint(xc,y,color);
   PlotPoint(xc,b,color);
  }
  else {
   PlotPoint(xc,y,image[CurImage][xc][y]);
   PlotPoint(xc,b,image[CurImage][xc][b]);
  }
  xc+=n;
 }

 n=((b-y)>0)?1:-1;

 for (t=0;t<=iplus(b-y);t++) {
  if (c) {
   if (final) {
    image[CurImage][x][yc]=color;
    image[CurImage][a][yc]=color;
   }
   PlotPoint(x,yc,color);
   PlotPoint(a,yc,color);
  }
  else {
   PlotPoint(x,yc,image[CurImage][x][yc]);
   PlotPoint(a,yc,image[CurImage][a][yc]);
  }
  yc+=n;
 }

 if (c)
  BLINEDRAWN=TRUE;
 else
  BLINEDRAWN=FALSE;
}

/*****************************************************\
 *                    MakeLine                       *
 *                                                   *
 *      erzeugt die fertige Linie im Mal-Feld        *
\*****************************************************/

MakeLine(x,y,a,b,c,final,color)
int x,y,a,b,c,final,color;
{
float cx,cy,xc,yc,n;
int t,draw=1,cnter=0;
```

Listing Part2.c

```c

 xc=(float)x;
 yc=(float)y;
 cx=(float)((a-x));
 cy=(float)((b-y));
 n=plus((plus(cx)>plus(cy))?cx:cy);

 if (n==0) n=(float)1;
 cx/=n;
 cy/=n;

 for (t=0;t<=(int)n;t++) {
  if (c) {
   if (final) image[CurImage][rn(xc)][rn(yc)]=color;
   PlotPoint(rn(xc),rn(yc),color);
  }
  else
   PlotPoint(rn(xc),rn(yc),image[CurImage][rn(xc)][rn(
yc)]);
  xc+=cx;
  yc+=cy;
 }

 if (c)
  BLINEDRAWN=TRUE;
 else
  BLINEDRAWN=FALSE;
}

/******************************************************\
 *              FinishRect / FinishLine               *
 *     Diese Funktionen beenden das Zeichnen          *
 *          eines Rechtecks / einer Linie             *
\******************************************************/

FinishRect()
{
 if (DRAWOK) {
  if (LINEDRAWN) {
   DrawRect(p1x,p1y,p2x,p2y);
   LINEDRAWN=FALSE;
  }

  if (ERASE)
   MakeRect(p1x,p1y,p2x,p2y,1,1,0);
  else
   MakeRect(p1x,p1y,p2x,p2y,1,1,CurCol);

  DRAWOK=FALSE;
 }
}

FinishLine()
{
 if (DRAWOK) {

  if (LINEDRAWN) {
   DrawLine(p1x,p1y,p2x,p2y);
   LINEDRAWN=FALSE;
  }

  if (ERASE)
   MakeLine(p1x,p1y,p2x,p2y,1,1,0);
  else
   MakeLine(p1x,p1y,p2x,p2y,1,1,CurCol);

  DRAWOK=FALSE;
 }
}

/******************************************************\
 *                    convpos                         *
 *        Umrechnen von internen Positionen           *
 *               in Mal-Positionen                    *
\******************************************************/

convpos(x,y,a,b)
int x,y;
long *a,*b;
{
 *a=(long)((x*20)+65);
 *b=(long)((y*10)+23);
}

UBYTE TempIm[18][18];

/******************************************************\
 *                       Fill                         *
 *                                                    *
 * Füllen eines Bereichs mit einer Farbe (Floodfill) *
\******************************************************/

Fill(x,y,c)
int x,y,c;
{
int oc, /* alte Farbe */

 SET=1,
 a,b, /* Lauf-Variablen */
     lx,rx,ty,by; /* Ränder */


     for (a=0;a<18;a++)
      TempIm[a][0]=127,
      TempIm[a][17]=127,
      TempIm[0][a]=127,
      TempIm[17][a]=127;

     conv(x,y,&x,&y);
 oc=(int)image[CurImage][x][y];
 if (oc==c) return;

     for (b=0;b<16;b++)
      for (a=0;a<16;a++)
       undo[a][b]=image[CurImage][a][b],
       TempIm[a+1][b+1]=image[CurImage][a][b];

 x++;
 y++;

 TempIm[x][y]=128; /* Startpunkt */

 while (SET!=0) {
  SET=0;
  for (b=1;b<17;b++) {
   for (a=1;a<17;a++) {
    if (TempIm[a][b]==128) {
     ChkNSet(&(TempIm[a+1][b]),oc);
     ChkNSet(&(TempIm[a-1][b]),oc);
     ChkNSet(&(TempIm[a][b+1]),oc);
     ChkNSet(&(TempIm[a][b-1]),oc);
     TempIm[a][b]=c;
     PlotPoint(a-1,b-1,c);
     SET++;
    }
   }
  }
 }

     for (b=0;b<16;b++)
  for (a=0;a<16;a++)
       image[CurImage][a][b]=TempIm[a+1][b+1];

     DRAWIT=FALSE;
}

/******************************************************\
 *                     ChkNSet                        *
 *                                                    *
 *  Überprüft, ob Punkt die zu füllende Farbe hat,    *
 *     und setzt ihn auf die Arbeits-Farbe (128)      *
\******************************************************/

ChkNSet(i,o)
UBYTE *i;
int o;
{
 if (*i==o) *i=128;
}

/******************************************************\
 *                   DisplayMode                      *
 *                                                    *
 *                Zeigt einen Text an                 *
\******************************************************/

DisplayMode(md)
char *md;
{
 SetAPen(rp,15L);

 Move(rp,10L,214L);
 Text(rp,"Mode:",5L);
 Text(rp,md,6L);

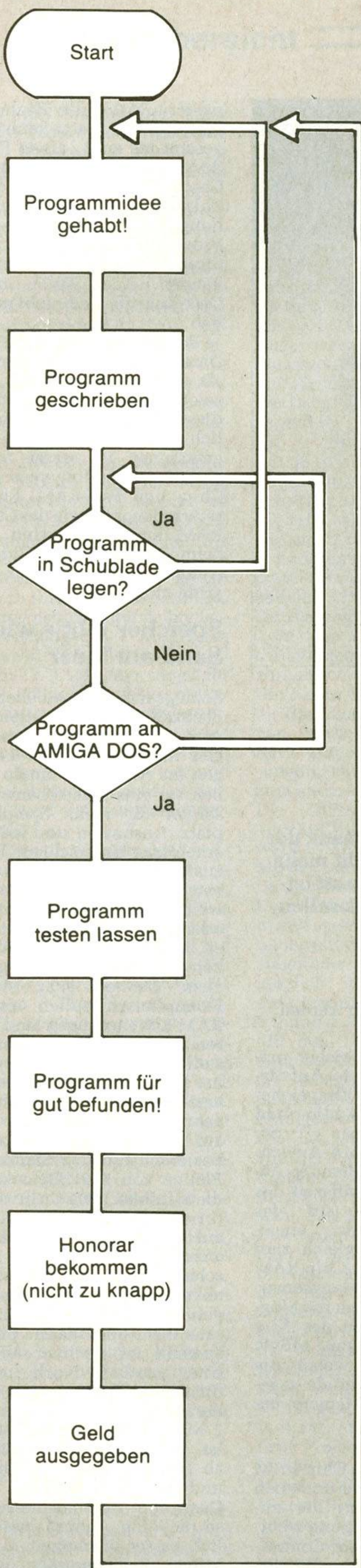
 SetAPen(rp,CurCol);
}

/******************************************************\
 *                       conv                         *
 *            wandelt die Maus-Position               *
 *        auf interne Werte (Mal-Fläche) um.          *
\******************************************************/

conv(x,y,a,b)
int x,y;
int *a,*b;
{
 x-=55;
 y-=18;
 *a=x/20;
 *b=y/10;

 if (*a>15) *a=15;
 if (*b>15) *b=15;
 if (*a<0) *a=0;
 if (*b<0) *b=0;
}
```

Listing Part2.c



VERSTEHEN SIE DAS?

Programmieren Sie für die AMIGA DOS !!

Haben Sie ein Super-Utility, die ultimative Dateiverwaltung oder den tiefblickendsten Disketten-Monitor für den Commodore Amiga geschrieben? Liegt Ihr Meisterwerk nun unbekannt und ungebraucht in der Schublade? Dann hätten wir einen Vorschlag für Sie!

Senden Sie Ihr bestes Programm doch einmal an die Redaktion der AMIGA DOS ein. Denn wenn uns Ihr Programm ebenso gut gefällt wie Ihnen, dann können Sie es kurz darauf in der AMIGA DOS gedruckt wiederfinden. **Gegen ein entsprechendes Honorar, versteht sich!** Und wenn uns Ihr Programm sogar noch besser gefällt als Ihnen selbst, dann hat es die gute Chance, **Programm des Monats** und mit

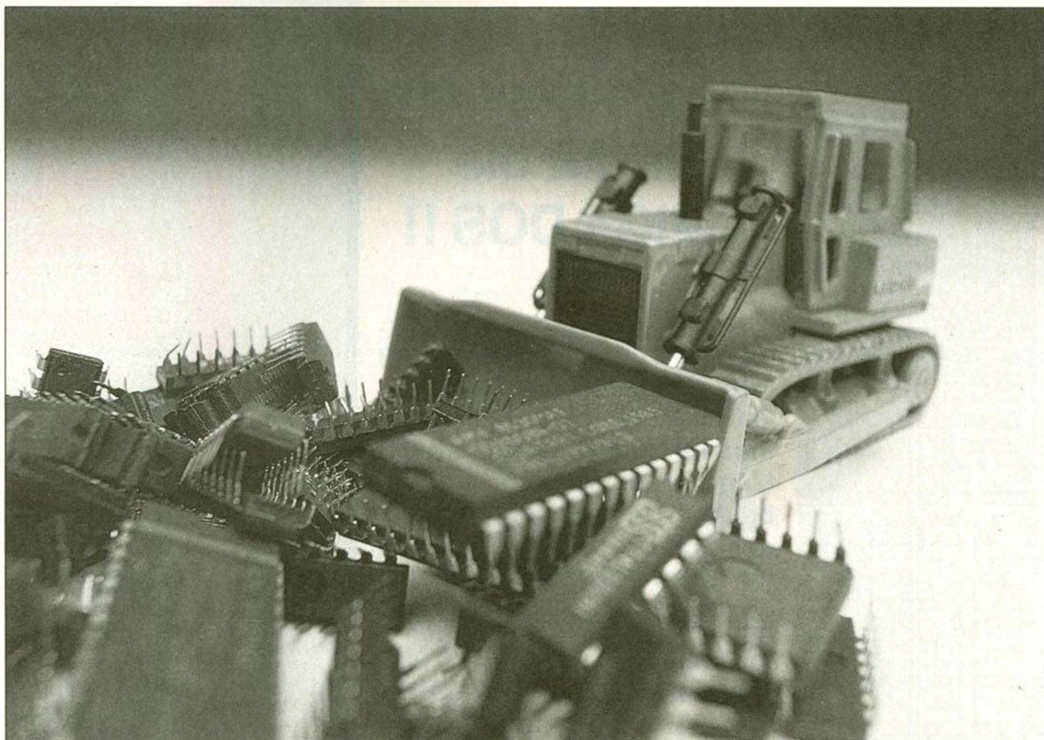
1500, – DM

prämiert zu werden. Also ran an die Maschine und programmiert. Oder an die Schublade und nachgeschaut. Es lohnt sich! Bestimmt!!

Ach ja – Einsenden können Sie es an den

DMV- Verlag
Redaktion AMIGA DOS
Postfach 250
3440 Eschwege

Auf geht's!!



Speicher satt?

RAM-Erweiterungen in der Übersicht

Das alte Klischee "Amiga, der reine Spielecomputer" gehört Gott sei Dank der Vergangenheit an. Wer ein richtiger "Amiganer" ist, begnügt sich nicht mehr mit dem relativ geringen Speicherplatz seines Amiga. Nein, Speicher satt ist angesagt! Sei es eine Festplatte, Filecard oder Speichererweiterung, es sollen schon einige Bit mehr in den Speicher passen.

Neuentwicklungen auf dem Softwaremarkt benötigen meist mehr Speicher, als die Grundausstattung des Amiga hergibt. Was also tun, wenn man beispielsweise mit DPaint III arbeiten möchte? Woher den nötigen Hauptspeicherplatz nehmen? Eine RAM-Erweiterung muß her, damit auch die großen Programme und Datenmengen ohne Einschränkungen genutzt werden können. Das Angebot auf dem Markt ist bei Speichererweiterungen für den Amiga sehr breit. Es reicht von 512-kByte-RAM-Erweiterungen für den A 500 bis zu 8-MByte-RAM-Erweiterungen für den A 2000.

Leistungsfähigkeit?

Generell sollte zunächst einmal die Frage geklärt werden: Was muß eine Speichererweiterung leisten, damit der Anwender zufrieden ist? Anhand dieser Fragestellung lassen sich folgende Kriterien aufstellen:

- benutzerfreundliche Autokonfigurierung
- keine Einschränkung der Arbeitsgeschwindigkeit
- stromsparende Chips (die meisten Platinen verwenden heute bereits 1-MBit-Chips)
- erweiterbare RAM-Karten, das heißt, diese Karten sollten noch weiter aufrüstbar sein
- Kompatibilität mit anderen Peripheriegeräten, die beispielsweise am Expansion-Port angeschlossen sind (Harddisk-Treiber etc.)

Speichererweiterungen sorgen nicht nur für mehr Platz im Speicher, sondern auch für höhere Geschwindigkeit, falls beispielsweise ein Programm im Fast-RAM abläuft. Generell sollte zunächst einmal zwischen zwei grundlegenden Speicherkonzeptionen unterschieden werden:

– Chip-RAM:
Dieser interne Speicher umfaßt bis zu 512 kByte. Auf diesen Direktzugriffsspeicher greifen Video-, Audio- und Disk-Anwendungen zu. Durch diese DMA-Anwendungen (Direct Memory Access) ist die Zugriffszeit des 68000-Prozessors auf das Chip-RAM stark eingeschränkt. Im Vergleich zum Fast-RAM ist das Chip-RAM nahezu 80 Prozent langsamer. Es besteht aber auch die Möglichkeit, mit Hilfe des "Big Agnus"-Chips bis zu 1 MByte zu adressieren, wobei die zweiten 512 kByte als sogenanntes Ranger-Memory bezeichnet werden.

– Fast-RAM:
Im Gegensatz zum Chip-RAM handelt es sich hier um einen externen Speicher, der zusätzlich zur Verfügung steht. Er besitzt eine eigene Controller-Logik und wird in den Erweiterungsslot des Rechners gesteckt. Der Amiga kann insgesamt bis zu 8 MByte Fast-RAM verwalten. In puncto Geschwindigkeit ist er dem Chip-RAM in jeder Beziehung überlegen, da weder Video- noch Audio-Chips auf diesen Speicher zugreifen können. Auch DMA- oder Disk-Zugriffe sind nicht möglich.

– Ranger-Memory:
Dieser Speicher könnte auch als internes Fast-RAM bezeichnet werden, jedoch trifft diese Beschreibung eigentlich nicht den Punkt. Genau genommen ist dieser Speicher ein Mittelding zwischen Chip- und Fast-RAM. Einerseits so langsam wie das Chip-RAM, andererseits sind wie beim Fast-RAM keine Video-, Audio-, DMA- und Disk-Zugriffe möglich.

Speicher gibt's wie Sand am Meer

Schlägt man eine x-beliebige Fachzeitschrift auf, fallen einem die vielfältigen Anzeigen für Speichererweiterungen ins Auge. Im Prinzip haben sie eins gemeinsam: Sie sorgen für mehr Speicherplatz. Ansonsten sind jedoch erhebliche Unterschiede festzustellen. So lassen sich die verschiedensten Speichererweiterungen entweder intern oder extern ein- bzw. anbauen, auch die Leistungsfähigkeit differiert bei nahezu gleichen Geräten sehr stark. Exemplarisch sollen einige RAM-Erweiterungen und deren Besonderheiten vorgestellt werden. Für den A 500 hat die Firma Gigatron eine variable Karte auf den Markt gebracht, die den 500er bis auf 1,8 MByte aufrüsten kann. Basierend auf einer Standard-Platine von 512 kByte kann die MiniMax 500 schrittweise (in 512-kByte-Schritten) bis auf 1,8 MByte aufgerüstet werden. Diese Karte ist abschaltbar und wird mit einer akkugepufferten Echtzeituhr geliefert. Die MiniMax ist abschaltbar und kann platzsparend im Rechner eingebaut werden. Noch mehr Speicher bekommt man mit der MiniMax Plus, die bis auf 2 MByte Speicher aufrüstbar ist. Sie verfügt serienmäßig ab 1 MByte über einen Gary- und einen CPU-Adapter.
Damit läßt sich beispielsweise der "Big Agnus" wesentlich besser ausnutzen. Ab 1 MByte Speichergröße wird

die MiniMax Plus mit einer RAM-Test-Diskette ausgeliefert. Daneben ist diese Karte noch resetfest und arbeitet mit Festplatten zusammen. Beide Karten sind autokonfigurierend und lassen sich relativ leicht einbauen. Auch die RAM-Erweiterung I und II für den Amiga 500, die die Firma HK-Computer vertreibt, kann auf den FAT bzw. BIG AGNUS zugreifen. Beide Erweiterungen sind mit 512 kByte Speicher ausgerüstet, wobei die RAM-Erweiterung II besonders stromsparend arbeitet.

Eine Erweiterung besonderer Art kommt von der Firma 3-State. Die A502 ist eine interne 512-kByte-RAM-Erweiterung, die es in sich hat. Sie überzeugt bereits durch ihre "Größe" – genauer gesagt durch ihre geringen Ausmaße – und natürlich durch ihre Leistungsfähigkeit. Ausgefeilte Megabit-Technologie und schnelle RAM-Chips sorgen für die nötige Zugriffsgeschwindigkeit und geringen Stromverbrauch. Die A502 erweitert den internen Speicher um 1 MByte, ist abschaltbar, kann mit und ohne Echtzeituhr geliefert werden und ist autokonfigurierend.

Im Bereich der externen Erweiterungen bietet die Firma Combitec die DRAM-Erweiterung für den A 500/1000 an. Diese dynamische Erweiterung kann den Amiga stufenweise auf 2, 4 oder 8 MByte aufrüsten. Ein DRAM-Controller sorgt für die nötige Zugriffsgeschwindigkeit. Zusätzlich verfügt diese Erweiterung über einen Festplattenadapter mit dem eine Harddisk der gleichen Firma angeschlossen werden kann. Ein kleines Manko soll nicht verschwiegen werden: Diese Erweiterung beansprucht einiges an Platz.

Von der Firma Kupke ist die Golem-RAM-Box für den A 500/1000 im Handel erhältlich, die am Expansionsport anzuschließen ist. Zum Lieferumfang gehören einige nützliche Utilities. Die RAM-Box ist in der Lage, den Speicher auf 2 MByte aufzurüsten, sie benötigt relativ viel Platz, ist abschaltbar und ab der Kickstart-Version 1.2 autokonfigurierend.

Nicht nur für die "Kleinen"...

...gibt es mehr Speicher. Kommen wir nun zum "Mini Flaggschiff", dem A 2000. Die Palette der RAM-Erweiterungen gestaltet sich auch für dieses Produkt recht vielfältig. Die CA2000.01 der Firma Jochheim stellt sich als zu allen Festplatten kompatibles Produkt vor. Diese Speichererweiterung ist jeder Zeit unter Verwendung eines neuen PAL-Satzes bis auf 8 MByte aufrüstbar. In der Grundausstattung kann sie wahlweise mit 2, 4, 6 oder 8 MByte bestückt werden. Die Autokonfiguration kann über einen Jumper abgeschaltet werden. Der Datenbus ist bidirektional gepuffert.

Die Speichererweiterung A2058 wird von der Firma Commodore angeboten. Schrittweise kann der A 2000 bis auf 8 MByte Speicher aufgerüstet werden. In der Grundausstattung wird diese Erweiterung mit 2 MByte ausgeliefert. Bei einer weiteren Aufrüstung sind Schritte von 2, 4 oder 8 MByte möglich, die nötigen Stecksockel sind vorhanden.

Die Firma CompuStore bietet die Speichererweiterung 8-Up! von MicroBotics an. Diese Karte gibt es in zwei Ausführungen, zum einen als DIP-, zum anderen als SIMM-Karte. Die DIP-Karte kann mit 1-MBit-Chips schrittweise (jeweils 2-MByte-Schritte) auf 8 MByte aufgerüstet werden. Demgegenüber kann die SIMM-Karte durch die Aufspaltung in zwei getrennte Speicherbereiche auf verschiedene Arten aufgerüstet werden. So ist es möglich, mit 256-kBit-Chips eine Erweiterung auf 0,5, 1, 1,5 und 2 MByte zu erzielen. Mit den 1-MBit-Chips kann dann um 2, 4, 6 und 8 MByte der Speicher erweitert werden. Somit wird eine sehr flexible Konfigurierung ermöglicht.

Die DRAM-2000-Karte wird von der Firma Combitec vertrieben. Es handelt sich bei dieser Karte um eine dynamische Fast-RAM-Erweiterung. Der Amiga 2000 kann um 2, 4 oder 8 MByte mit 1-MByte-SIP-Modulen aufgerüstet werden. Ein integrierter DRAM-Controller sorgt für null Waitstates. Zusätzlich gehört noch die Workbench 1.3 zum Lieferumfang. Bei der Aufrüstung selbst ist die Jumperstellung zu verändern.

Weitere Kurzinformationen über Preise und Leistungsfähigkeit der einzelnen auf dem Markt erhältlichen Produkte können Sie der Übersicht, die übrigens keinen Anspruch auf Vollständigkeit erhebt, entnehmen. (vb)

Anbieter	Typ	500	1000	2000	Preis	Extras
Alcomp GmbH Glescher Weg 22 5012 Bedburg Tel. 02272/2093	2/4 MByte RAM-Erweiterung	• •	– –	– –	DM 398,- DM 898,- DM 1298,-	teilbestückt ohne RAMs mit 2 MByte bestückt mit 4 MByte bestückt voll kompatibel, autokonfigurierend, einfacher Einbau ohne Löten
	8 MByte RAM-Erweiterung	• • • •	• • • •	• • • •	DM 698,- DM 998,- DM 1598,- DM 2498,-	Erweiterung ohne RAMs Mit 2 MByte bestückt Mit 4 MByte bestückt Mit 8 MByte bestückt Frei bestückbar von 512 kByte bis 8 MByte mit 41256 u. 511000 RAMs, autokonfigurierend, abschaltbar, durchgeschleifter Bus und Metallgehäuse bei A 500/1000
	512 kByte RAM-Erweiterung	•	–	–	a. Anfrage	autokonfigurierend, abschaltbar, Uhrenschaltung auf Platine nachrüstbar
BSC GmbH Schleißheimer Str. 205a 8000 München 40 Tel. 089/3084152	EXP 1000	•	–	–	DM 589,-	unbestückt
	EXP 1000/1	•	–	–	DM 948,-	1 MByte bestückt interne Speichererweiterungskarte autokonfigurierend, in Kombination mit Standard-RAM-Erweiterung bis zu 2 MByte Fast-RAM
	EXP 8000 EXP 8000/2 EXP 8000/4 EXP 8000/6 EXP 8000/8	• • • • •	– – – – –	– – – – –	DM 878,- DM 1460,- DM 1990,- DM 2460,- DM 2850,-	unbestückt 2 MByte bestückt 4 MByte bestückt 6 MByte bestückt 8 MByte bestückt interne autokonfigurierende Speichererweiterungskarte, Mega-Bit-Chips
	PRORAM 2000 PRORAM 2000/2 PRORAM 2000/4 PRORAM 2000/6 PRORAM 2000/8	– – – – –	– – – – –	• • • • •	DM 539,- DM 1160,- DM 1730,- DM 2230,- DM 2650,-	unbestückt 2 MByte bestückt 4 MByte bestückt 6 MByte bestückt 8 MByte bestückt interne autokonfigurierende Speichererweiterungskarte, Mega-Bit-Chips
Combitec GmbH Liegnitzer Str. 6-6a 5810 Witten Tel. 02302/88072	DRAM 512	•	–	–	DM 169,-	intern, Anschlußmöglichkeiten für Abschalter, ohne Uhr
	DRAM 2/4/8 MByte	•	•	–	DM 998,- DM 1798,- DM 2998,-	mit 2 MByte bestückt (Aufpreis für A 1000 mit 4 MByte bestückt je DM 80,-) mit 8 MByte bestückt durchgeführter Bus, (bei A 1000 Version mit Treiberbausteinen), abschaltbar, die kleineren Versionen sind intern aufrüstbar, integrierter Adapter für Festplatte HD 20/40
	MultiMegaCard	–	–	•	DM 379,- DM 979,- DM 1698,- DM 2798,-	0 MByte mit 2 MByte bestückt mit 4 MByte bestückt mit 8 MByte bestückt optional mit SIP-Modulen oder Megabitchips bestückbar, autokonfigurierend, keine Waitstates, asynchrone Taktfrequenz
Compu Store GmbH Fritz-Reuter-Str. 6 6000 Frankfurt 1 Tel. 069/567399	Null-kByte-8-MByte-Karte (DIP/SIMM)	–	–	•	DM 600,-	Speicherkarte um je 2 MByte aufrüstbar, DIP-Karte wird mit 1-MBit-Chips, die SIMM-Karte mit 512-kByte-Chips und 1-MByte-Chips aufgerüstet werden, autokonfigurierend, keine Waitstates, flexible Konfigurierung
	2-MByte-8MByte-Karte (DIP)	–	–	•	DM 1000,-	o.a. Karte mit 2 MByte aufgerüstet, bis auf 8 MByte aufrüstbar
	2 MByte RAM-Chips	–	–	•	DM 450,-	16 Chips (1 MBit)
EUROSYSTEMS Hühnerstr. 11 4240 Emmerich Tel. 02822/45589 u. 45923	512 kByte RAM-Erweiterung	•	–	–	DM 69,- DM 109,-	ohne RAMs, abschaltbar inklusive Uhr u. Kalender (RAMs auf Anfrage)
Gigatron Resthauser Str. 128 4590 Cloppenburg Tel. 04471/3070 u. 83740	MiniMax 500	•	–	–	DM 288,- DM 508,- DM 648,- DM 788,-	ohne Aufrüstsatz, 512 kByte bestückt mit 1 MByte bestückt mit 1,5 MByte bestückt mit 1,8 MByte bestückt inklusive akkugepufferte Echtzeituhr, erweiterbar, abschaltbar
	MiniMaxPlus	•	–	–	DM 288,- DM 568,- DM 708,- DM 848,-	ohne Aufrüstsatz, mit 1 MByte bestückt mit 1,5 MByte bestückt mit 2 MByte bestückt erweiterbar, inklusive akkugepufferte Echtzeituhr, ab 1 MByte inklusive Gary- & CPU-Adapter, RAM-Test-Diskette, unterstützt "Big Agnus"
	Amiga 2000	–	–	•	DM 698,- DM 898,-	bestückt mit 1 MByte bestückt mit 2 MByte interne Speichererweiterung, bis 8 MByte aufrüstbar (in 0,5-MByte-Schritten bis 6 MByte, + 2 MByte bis 8 MByte)
HK-Computer Bonner Str. 37 5000 Köln 1 Tel. 0221/311606	RAM-Erweiterung I	•	–	–	DM 189,-	inkl. Uhr, abschaltbar, 512 kByte, läuft problemlos mit BIG/FAT AGNUS
	RAM-Erweiterung II	•	–	–	DM 198,-	Megabittechnologie, 512 kByte, mit Uhr, abschaltbar, weniger Stromverbrauch, läuft problemlos mit BIG/FAT AGNUS
	RAM-Erweiterung	•	–	–	DM 698,-	1,8 MByte, mit Uhr, autokonfigurierend, aufrüstbar, mit Gary-Adapter, schnelle Zugriffszeiten, bei 1 MByte Chip-Memory werden nur 512 kByte erkannt, da EXRAM-Leitung hochgelegt, läuft problemlos mit BIG/FAT AGNUS
	RAM-Erweiterung	–	–	•	DM 848,-	mit 2 MByte bestückt, bis auf 8 MByte durch Umstecken zusätzlicher RAMs und Jumpers aufrüstbar, autokonfigurierend, lieferbar mit 0/2/4/6/8 MByte, pro 2 MByte werden 16 RAMs benötigt, läuft ohne Probleme mit BIG/FAT AGNUS
H. Schomburg Anna Seghers Str. 99 2800 Bremen 61 Tel. 0421/833864 u. 388667	512 kByte RAM	•	–	–	DM 209,-	RAM-Karte intern resetfeste Uhr abschaltbar
	1,8 MByte Intern	•	–	–	DM 775,-	Akkuuhr, RAM-Test-Disk abschaltbar

Anbieter	Typ	500	1000	2000	Preis	Extras
H. Schomburg Anna Seghers Str. 99 2800 Bremen 61 Tel.: 0421/833864 u.388667	2 MByte RAM-Karte	–	–	●	DM 899,-	8-MByte-Karte mit 2 MByte bestückt keine Waitstates 100 ns RAMs Megabitchips
	Animate-Turbo-Board IIa	●	●	●	ab DM 849,-	mit 68020, 68881 bestückt (16MHz) umschaltbar auf 68000 auch für Amiga mit MC68882 (20MHz)
	Animate-Turbo-Board III	–	–	●	ab DM 1350,-	mit 68020, Slotkarte für A 2000 mit 1 MByte 32-Bit-RAM bestückbar auch für Amiga mit MC68882 (20MHz)
Intelligent Memory Wächtersbacherstr. 89 6000 Frankfurt 61 Tel. 069/410071 u. 410072	IM ProRAM	●	–	–	DM 159,- DM 299,- DM 449,- DM 599,- DM 749,-	0 MByte mit 0,5 MByte bestückt mit 1 MByte bestückt mit 1,5 MByte bestückt mit 1,8 MByte bestückt kompatibel, abschaltbar, bestückt mit 80ns-RAMs, 1,5/1,8 MByte auch unter Kickstart 1.2
Jochheim Ralf Computer Tuning Osnabrücker Str. 96 4802 Halle Tel. 02823/1275	CA 2000.01	–	–	●	DM 898,- DM 1398,- DM 1898,- DM 2398,-	mit 2 MByte bestückt mit 4 MByte bestückt mit 6 MByte bestückt mit 8 MByte bestückt Autokonfigurierung (ist abschaltbar), Datenbus bidirektional gepuffert, Refreshcontroller durch PAL-Realisierung, zu den meisten Festplatten kompatibel
Kupke Computertechnik Burgweg 52a 4600 Dortmund Tel. 0231/818325-27	Golem 512-kByte-Karte	●	–	–	DM 249,-	mit gepufferter Uhr, abschaltbar, mit stromsparender Megabittechnologie autokonfigurierend
	Golem RAM-Box	●	●	–	DM 998,-	externe RAM-Box für A 500/1000, abschaltbar, autokonfigurierend, bis auf 8 MByte erweiterbar
	8-MByte-Steckkarte	–	–	●	DM 998,-	mit 2 MByte bestückt, bis auf 8 MByte aufrüstbar, echtes Fast-RAM nach Golem Standard
Rainbow Data Am Kalkofen 32 5603 Wülfrath Tel. 02058/1366	512 kByte RAM-Erweiterung	●	–	–	DM 219,-	Uhr abschaltbar
	1,8 MByte RAM	●	–	–	DM 798,-	Uhr abschaltbar
	2 MByte Profex SE 2000	–	–	●	DM 798,-	Bus abschaltbar
	2 MByte Box Extern	●	●	–	a. Anfrage	auch teilbestückt (512 kByte u. 1 MByte)
	4 MByte Box Extern A 1000 A 2000	–	●	●	a. Anfrage	durchgeführter Bus 2 MByte u. 4 MByte bestückt 8 MByte u. 2 MByte bestückt
Roßmöller Computer Technik Neuer Markt 21 5309 Meckenheim Tel. 02225/2061-63	A512	●	–	–	DM 179,-	512-kByte-Speichererweiterung, inklusive Uhr, abschaltbar
	A2MByte	●	–	–	DM 598,-	2-MByte-RAM-Karte läuft mit FAT/BIG AGNUS
	A8MByte/1000	–	●	–	DM 798,-	8-MByte-RAM-Karte, mit 2 MByte bestückt, abschaltbar, keine Waitstates
	A8MByte/2000	–	–	●	DM 698,- DM 1998,-	8-MByte-RAM-Karte mit 2 MByte bestückt mit 8 MByte bestückt 4-MBit-Technologie, autokonfigurierend, keine Waitstates, abschaltbar
Supra Corporation Oltersdorf EDV Kurzestr. 20 4352 Herten 6 Tel. 0209/620408	Supra RAM 2000	–	–	●	DM 1098,-	mit 2 MByte bestückt, lieferbar in jeder Ausbaustufe 0/2/4/6/8 MByte, kompatibel zu anderen Erweiterungen (sowie zu 512 kByte und 1 MByte Chip-RAM-Geräten), Megabit-RAMs, inklusive Testsoftware
Tröps & Hierl Jordanstr. 3 5040 Brühl Tel. 02232/45018	ERAM-MEGA	●	–	–	DM 199,-	512-kByte-Erweiterung Megabitspeicher, geringer Stromverbrauch, akkugepufferte Echtzeituhr, erweiterbar
	MEGA-MODUL	●	–	–	DM 250,- DM 375,- DM 489,-	Erweiterung auf 1 MByte Erweiterung auf 1,5 MByte Erweiterung auf 1,8 MByte modulare Erweiterung, stufenweise aufrüstbar bis 1,8 MByte, intern einbaubar, ansonsten wie ERAM-MEGA
Vesalia Computer Magdalenenweg 4 4230 Wesel Tel. 0281/65466	Winner-RAM-Karte	●	–	–	DM 198,-	512-kByte-Erweiterung, abschaltbar, inklusive Uhr, zirka 20% Stromverbrauch
	512-kByte-Karte	●	–	–	DM 169,-	komplett bestückt, abschaltbar
	1,8-MByte-Karte	●	–	–	DM 698,-	komplett bestückt, abschaltbar, inklusive Uhr
	2-MByte-Box 2-MByte-Box 2-MByte-Box 4-MByte-Box	– – – –	● ● ● ●	– – – –	DM 498,- DM 798,- DM 1148,- DM 1588,-	512 kByte bestückt mit Bus mit Busdurchführung komplett mit Uhr und Kickstart 1.3 mit Busdurchführung
	8-MByte-Karte	–	–	●	DM 948,- DM 1680,- DM 2948,-	mit 2 MByte bestückt mit 4 MByte bestückt mit 8 MByte bestückt
3-State Schaumburgstr. 17 4350 Recklinghausen Tel. 02361/492928 u. 16207	A 502	●	–	–	DM 158,- DM 178,-	ohne Uhr erweitert Speicher bis auf 1 MByte, inkl. Uhr, autokonfigurierend, abschaltbar, Mega-Bit-Technologie
	A 580	●	–	–	DM 328,- DM 428,- DM 698,-	mit 512 kByte bestückt mit 1 MByte bestückt mit 1,8 MByte bestückt inklusive Uhr, Akku, Gary-Adapter, autokonfigurierend, ab 1,5 MByte wird die Kickstart Version 1.3 vorausgesetzt
	A 1001	–	●	–	DM 398,-	Erweiterung auf 1 MByte, intern, läuft mit vielen Erweiterungen, abschaltbar
	CA 2000	–	–	●	DM 898,- DM 1398,- DM 1898,- DM 2398,-	mit 2 MByte bestückt mit 4 MByte bestückt mit 6 MByte bestückt mit 8 MByte bestückt autokonfigurierende RAM-Karte, keine Waitstates, schnelle Megabit-RAMs, abschaltbar

Die PD-Werkzeugkiste

Wer Anwendungsprogramme für seinen Amiga haben möchte, ohne viel Geld dafür ausgeben zu wollen, der sollte sich auf dem immer größer werdenden Public-Domain-Markt umsehen. Hier kann so manches Schnäppchen gemacht werden, das selbst kommerzielle Programme das Fürchten lehren kann.

Wir stellten in der letzten Ausgabe schon einige Disketten aus der Cactus-Reihe vor, die Sie sich besorgen sollten. Auch diesmal haben wir wieder etwas von dieser Gruppe zu berichten. Langer Rede, kurzer Sinn, stürzen wir uns gleich in die Fluten der PD-Programme auf der Cactus 28.

Elemente zum Anfassen und Verstehen

Mit dem Programm **Elements** sind besonders die Chemiefreunde angesprochen. Elements ist eine Demo-Version des Programms CAMM, das besonders auf die dreidimensionale Darstellung von Molekülen eingeht.

Das Programm wird aus dem CLI gestartet, worauf sich das Periodensystem der chemischen Elemente auf dem Bildschirm präsentiert. Durch Anwählen des jeweiligen Feldes mit der Maus kann man sich über alle Elemente kundig machen. Es öffnet sich dann ein Fenster, in dem die Eigenschaften des Atoms (Atomgewicht, Siede- und Schmelzpunkt, Elektronegativität, Kristallstruktur etc.) angezeigt werden. Wer sich in der Chemie noch wie auf brüchigem Eis bewegt, findet in einem Informationsmenü effektive Hilfe. Hier werden grundsätzliche Begriffe der Chemie verständlich erklärt – der Lernerfolg bleibt also nicht aus.

Bei dem Programm **GPRINT** handelt es sich um ein Hilfsprogramm, das zweifarbige IFF-Bilder auf einem Epson-kompatiblen Drucker ausdruckt. Dabei gestaltet sich GPRINT durch eine mögliche Zusammenarbeit mit Deluxe Paint um so interessanter, da sich mit diesem Grafikprogramm erstellte Bilder problemlos ausdrucken lassen.

Um eine hohe Flexibilität zu garantieren, bietet das Programm neun verschiedene Druckarten, die sich aus vier Druckdichten und drei Druckqualitäten kombinieren lassen. Wer nun denkt, daß dies alles sei, was das Programm zu bieten hat, sieht sich glücklicherweise getäuscht. Durch die Eingabe einer "Zwei-Buchstaben-Kombination" läßt sich die Druckerauflösung definieren. Die Auflösung steht jedoch in Abhängigkeit zum Format der gewählten Hires-Grafik.

Wie aus dem Programmnamen ersichtlich wird, ermöglicht der Pointer-Animator eine Modifizierung des Mauszeigers. Es handelt sich jedoch nicht um einen gewöhnlichen Austausch des Pointer-Files. Die Datei wird nämlich durch eine Reihe von Bildern ersetzt, die in einem IFF/ILBM-File abgespeichert werden. Hierdurch wird die Sequenz der Bilder festgelegt, so daß der Eindruck einer fortlaufenden Bewegung entsteht. So kann es passieren, daß sich ein fröhlicher Smily-Pointer langsam in einen blutrünstigen Halloween-Kürbis verwandelt. Zusätzlich kann man anstatt der drei Farben nun 15 Farben in einem Pointer darstellen. Doch das ist noch nicht alles, denn weiterhin läßt sich auch die Ablaufgeschwindigkeit der Bilder sowie die Größe des Pointers bei der Erstellung problemlos variieren.

Epidemie Virus contra Amiga – die häufigste Computerkrankheit

Die Virenverseuchung beim Amiga ist immer wieder ein akutes Problem. Wie allgemein bekannt, setzen sich die meisten Viren auf den Bootsektoren fest, von denen sie sich unaufhaltsam verbreiten und Schaden anrichten. Dagegen gibt es Virenjäger, die die Bootsektoren überprüfen und gegebenenfalls desinfizieren. Jedoch schützen diese Programme keine kommerziellen Disketten (z.B. bei Spielen), auf deren Bootsektoren sich meist Schnellader oder Routinen befinden, die das Betriebssystem ausschalten.

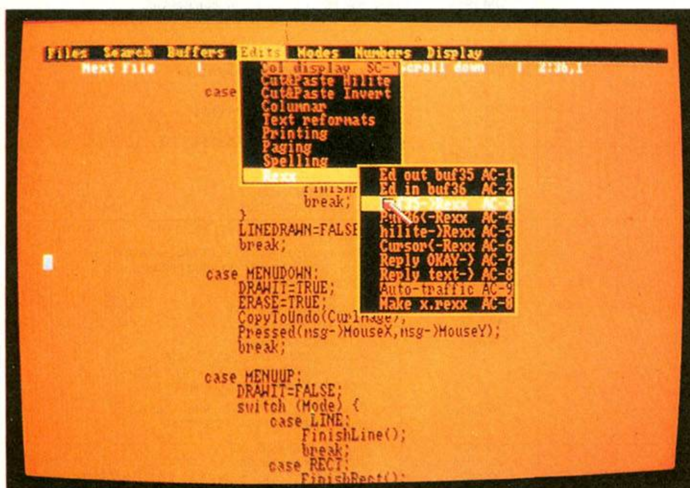
Wenn einmal der Bootsektor verseucht ist, kann die ganze Diskette als wertlos bezeichnet werden. Die Disketten können dann nur noch formatiert und als Datendiskette benutzt werden – teure Leerdisketten? Nicht unbedingt, denn das Programm **SafeBoot** soll bei diesem Übel Abhilfe schaffen. Dazu liest SafeBoot die Bootsektoren der gewünschten Diskette ein, um sie dann auf einer anderen Diskette wieder abzuspeichern. Falls die Bootsektoren Ihrer Originaldiskette von den Viren befallen sind, ersetzt SafeBoot die befallenen Bootblöcke durch die zuvor abgespeicherten Bootsektoren. Die Voraussetzung ist natürlich, daß die abgespeicherten Bootsektoren virenfrei sind.

Wer Ordnung hält, ist nur zu faul zum Suchen...

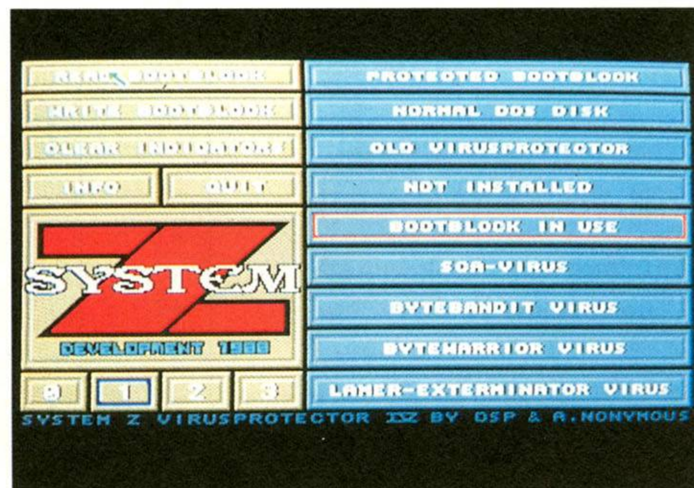
Mit **LS** präsentiert sich ein überarbeiteter Filelister, der vor allem die DOS-Befehle List und Dir ersetzen soll. Bei Eingabe der Dateinamen können auch Platzhalter, sogenannte Wildcards, verwendet werden. Des weiteren verfügt LS über Sortierfunktionen. Durch bestimmte Optionen werden die Dateien nach verschiedenen Kriterien sortiert. So können die Files nach Länge oder nach Erstellungsdatum geordnet werden.

(Jürgen Seibel/br)

Name: Cactus 28
Quelle: APS Electronic



Uedit, ein Texteditor der Extraklasse, kann mit einer Menge Funktionen aufwarten



Der System Z Virusprotector IV macht nun auch mit dem Lamer-Exterminator-Virus Schluß

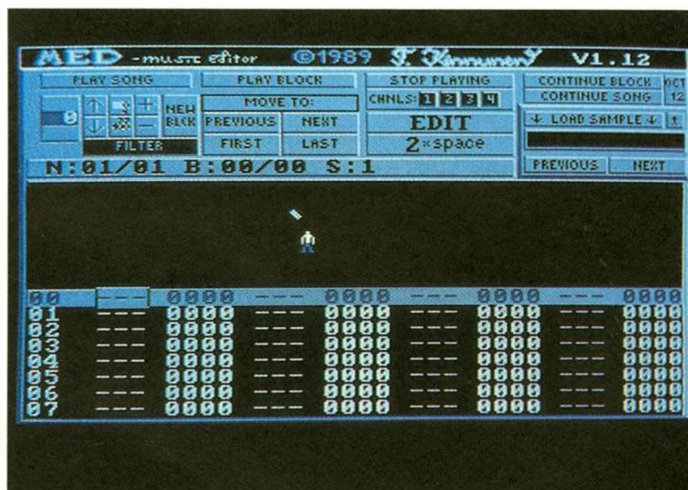
Mit dem Programm **Show** lassen sich IFF-Bilder auf dem Bildschirm darstellen. Dabei verbraucht das Programm nur wenig Speicherplatz, so daß es auf jeder Diskette Platz findet.
Hinter dem dubiosen Namen **MEINCLI** verbirgt sich ein erweiterter Ersatz des Command Line Interpreters der Workbench 1.2. Zur Steigerung der Geschwindigkeit wurden die Hauptbefehle des CLI in das RAM gelegt. Zudem haben die Programmierer einige Befehle modifiziert. Eine weitere nützliche Option stellt der Befehl DEFINE dar. Durch ihn können Tasten auf einfachste Art und Weise belegt werden, die zudem noch abgespeichert werden können.
TSIZE ist ein sehr nützlicher Befehl, durch den die Länge eines Files in Bytes ermittelt wird.
Bei **CLED** handelt es sich um einen Kommando-Zeilen-Editor, ähnlich dem ed der Workbench.
Sonix-Player ist eine Abspielroutine für Musikstücke, die mit dem Musikprogramm Sonix erstellt wurden. Somit ist es auch möglich, Sounds abzuspielen, ohne daß sie auf das Hauptprogramm Sonix zugreifen müssen.
Wem ist es noch nicht passiert, daß sich ein Bekannter oder Verwandter für den optimalen Computerfreak hielt, und im Wahn seines Unwissens die Daten restlos zerstörte? Das Programm **Tastenschloß** soll den Benutzer vor diesem Unheil bewahren. Dazu friert das Programm die Maus und die Tastatur bis zur Eingabe eines gewählten Passworts ein. Dabei ist kein Cursor oder bewegender Pointer zu erkennen. Die Änderung des Passwortes erfordert etwas mehr Mühe: Zuerst lädt man ein Filezap-Programm, um darauf das Tastenschloß-File in den File-Zapper zu laden. Am Ende des Tastenschloßprogramms befindet sich das Passwort. Nachdem dieses geändert wurde, kann das neue Passwort mit zusätzlichen Parametern abgespeichert werden.
Das Programm **BigView** stellt Bilder, die über die normale Bildschirmgröße reichen, dar. Dies wird durch das Scrolling des überdimensionalen Bildes erreicht. Das Programm **Clock** zeigt einzig und allein durch seinen Aufruf die Uhrzeit und den freien Speicher an.
Wie es hereinschallt, so schallt es auch heraus. Eine Feststellung, die besonders bei dem Programm **Echo** zutrifft. Hier sieht man das Bemühen der Programmiererin in dem Programm bestätigt. Es handelt sich dabei um eine Modifizierung des Echo-Befehls aus dem CLI.
Nun ist es möglich – was unter dem Standard-Befehl nicht möglich war – das mit Echo dargestellte String zum Drucker zu schicken. Dabei werden durch zusätzliche Parameter die Features, wie Boldface, Italic oder Underline, genutzt. In einem Extra-Verzeichnis befindet sich eine Reihe von Spielen und ein Anwenderprogramm, die in Basic programmiert wurden. So bieten die Spiele Superhirn und Börsenspiel eine interessante Abwechslung. Mit dem BASIC-Programm **Kassettendruck** lassen sich Labels für Musikkassettenhüllen leicht selbst erstellen.
(Jürgen Seibel/br)

Name: Cactus 1
Quelle: APS Electronic

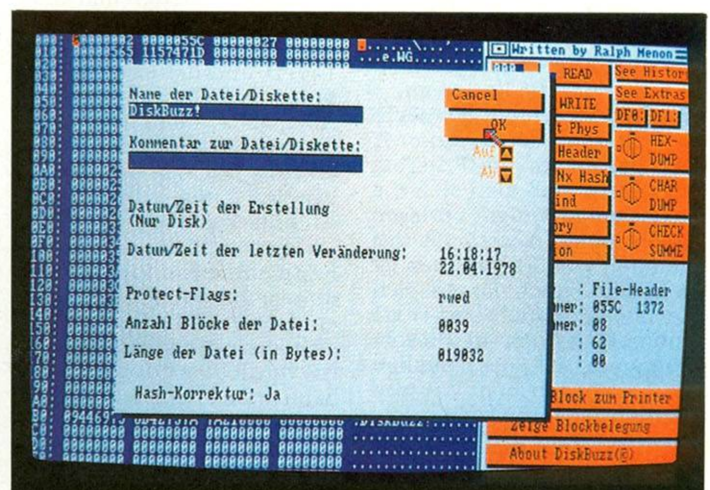
Die deutsche PD-Reihe Bavarian präsentiert auf der Diskette Numero 7 gleich eine ganze Serie von Texteditoren und Textverarbeitungen.

Editoren setzen Maßstäbe

DME heißt der erste Texteditor, der durch Benutzerfreundlichkeit auffällt. Nach dem Aufruf des Programms erscheint das Arbeitsfenster. Um einem erstellten Text ein professionelles Outfit zu verpassen, stehen dem Benutzer eine Vielzahl an Funktionen zur Verfügung, die man meist nur in kommerziellen Textverarbeitungen wiederfindet. Nehmen wir als Beispiel die 'Suchen und Ersetzen'-Option. Hier kann im gesamten Text nach einem frei definierten String (beliebige Zeichenkette) gesucht, und durch ein anderes frei wählbares String ersetzt werden.
Ein weiterer nützlicher Menüpunkt sind die Blockoperationen. Hierbei kann ein Textabschnitt markiert werden, mit dem dann verschiedene Operationen, wie zum Beispiel Kopieren, Einfügen oder Ausdrucken des gewählten Blocks, durchgeführt werden können. Wer gerne mit Makros arbeiten will, hat zudem die Möglichkeit, sich bestimmte Tasten mit Textteilen zu belegen. Dies geschieht mit dem Befehl Map.
Hinter dem Begriff Fensterln verbirgt sich nicht die bayrische Prozedur mit Leiter und Blumenstrauß, sondern er bedeutet das gleichzeitige Bearbeiten mehrerer Files. Einige Tastaturbelegungen, wie Top (Sprung an den Textanfang) oder Formatieren des Textes runden das positive Bild dieses Editors ab.
Mit **Hed** reiht sich ein weiterer Editor in den Bereich der Textbearbeitung bzw. Texterstellung ein.
Im Gegensatz zu dem obig beschriebenen Editor verfügt Hed nicht über die massive Anzahl der Optionen. HED zeichnet sich jedoch durch eine übersichtliche und vereinfachte Menügestaltung aus. Durch die Tastenbelegung nutzt das Programm auch Befehle wie Find, also die Stringsuche.
Uedit stellt den wohl komfortabelsten Editor auf dieser Diskette dar. Die Vorteile von HED und DME scheinen in diesem Editor vereinigt.
Nicht allein die Menügestaltung, sondern auch die Masse der Befehle prägen den als sehr gut zu bezeichnenden Editor. Das Hauptmenü selbst teilt sich in sieben Kategorien: Files, Search, Buffers, Edits, Modes, Numbers und Display.
Mit diesen Menüpunkten lassen sich unter anderem Blöcke markieren, Tasten belegen, Ausdrucke erstellen und Texte formatieren. Aber auch DOS-Kommandos können direkt aus dem Editor auf-



Soundtrackerähnlich zeigt sich MED, der Musikeditor auf der Time 5



Diskbuzz ist ein bedienungfreundlicher Diskettenmonitor, mit dem man auch editieren kann



Mit der Demoversion des Relo-Diskmasters lassen sich bequem Dateioperationen durchführen. Ideal für Anwender, die mit dem CLI auf Kriegsfuß stehen

Mathematische Funktionen grafisch auf den Monitor ausgegeben – kein Problem mit R.O.M.

gerufen werden. Dies sind nur einige der vielen Optionen dieses Editors, der es meiner Meinung nach mit vielen kommerziellen Editoren aufnehmen kann. Ein Tutorial macht den Einstieg in den Editor leichter.

Der Name gibt eigentlich über die Wirkung des Programms schon ausreichend Auskunft. Das primäre Ziel des Programms **Blitz** liegt nämlich darin, Textdateien möglichst schnell auf dem Bildschirm zu präsentieren. Dazu läuft Blitz auch im Hintergrund ab. Durch eine Tastenkombination kann das Programm jederzeit aktiviert werden. Es dient somit also mehr oder weniger als Alternative zu dem Type- und More-Befehl. Zusätzlich wartet Blitz mit einer Formatier-Option, Save-Routinen und einem blitzschnellen Scrolling auf.

Wenn man sich die auf der Diskette befindlichen Editoren ansieht, so kann schon fast ein Vergleich mit einer Olympiade gezogen werden: Es treten nur die Besten an.

PrinText V1.2 heißt der nächste Editor auf der Bavarian 7. Das Besondere an PrinText liegt darin, daß man sich die Texte vom Amiga vorlesen lassen kann. Aber auch das Einladen von IFF-Grafiken ist möglich, jedoch lassen sich die Grafiken nicht in Texte einbinden. Auch in diesem Programm ist das Suchen nach vorgegebenen Wörtern möglich.

(Jürgen Seibel/br)

Name: Bavarian Nr. 7
Quelle: Bavariansoft

Time nennt sich eine PD-Reihe, deren Diskette Nummer fünf besondere Beachtung verlangt. Brandneue Utilities (unter anderem VirusX 4.0 und System Z-Virusprotector IV) machen diese Diskette zum Helferlein der Amigabenutzer. Schauen wir uns den Inhalt doch einfach einmal etwas näher an.

Zeit ist Geld – Time 5

VirusX von Steve Tibbett kennt man ja schon seit einiger Zeit auf dem Amiga. Nun ist das Update erschienen, das den Viren den Garaus machen soll. VirusX sollte an erster Stelle der Startup-Sequence stehen, damit die nachfolgenden Befehle auf Link-Viren (Viren, die sich an bestimmte DOS-Befehle anhängen und nicht auf den Bootsektor zuzugreifen brauchen) untersucht werden können. Dabei werden der KickTagPrt und der CoolCapture-Vektor auf Viren abgescheckt. Aber auch ältere Viren werden nicht verschont: VirusX erkennt fast alle bisherig bekannten Viren auf dem Amiga, untersucht auch das RAM und eventuell angeschlossene Festplatten. Hat VirusX einen dieser bösen Kuckuckseier entdeckt, kann der Benutzer sich aussuchen, ob er den Virus vernichten oder ob er alles beim alten lassen will. Letzteres ist besonders dann anzuraten, wenn es sich um kommerzielle Software handelt, die den Bootblock für wichtige Daten reserviert. Auch wenn man seine Disketten mit einem Bootintro versehen hat, meldet sich VirusX mit der Meldung "Danger: The disk has Nonstandard Boot Code".

Zwei Jäger bilden eine unschlagbare Einheit gegen Viren

Der **System Z Virusprotector IV** stellt einen Bruder zu VirusX 4.0 dar. Die Benutzung ist zwar wesentlich bedienungsfreundlicher, als dies bei VirusX der Fall ist, jedoch erkennt dieses Programm nicht alle Viren. Es ist aber mit dieser Version auch möglich, die erste Version des Lamer Exterminators (Linkvirus) festzustellen. Dieser Virus äußert sich meist durch Read/Write-Errors und schreibt in manche Dateien unzählige Male das Wort Lamer! Dieser Virus sorgte in letzter Zeit bei vielen Amigabesitzern für Furore. War es doch schier unmöglich, den Virus ausfindig zu machen und zu vernichten, da er auch das RAM belegte und zudem noch selbstmodifizierend war. Wenn man die Verseuchung auf den Disketten feststellte, war es schon zu spät, die Diskette war wertlos geworden. So blieb als einziges Mittel zur Bekämpfung des Lamer-Virus das Formatieren der Diskette, was natürlich den bedingungslosen Datenverlust bedeutete.

Ein übersichtliches Menü des System Z-Virusprotectors IV ermöglicht das Einlesen des Bootblocks. Nun wird die Diskette auf den SCA-, Bytebandit-, Bytewarrior- und die Ur-Versionen des Lamer-Exterminator-Virus überprüft.

Falls ein Virus in die Klauen des Programms geraten sollte, wird er erbarmungslos beseitigt und der gereinigte Bootblock abgespeichert. Damit das Überprüfen auf Viren schneller abläuft, können bis zu drei Laufwerke angesprochen werden.

Nach dem Einladen der Workbench ertönte ein einschlägiger Sound aus den Lautsprechern unseres Monitors, der uns während der Beschäftigung mit der Workbench begleitete. Dieser Song wurde mit dem auf der Diskette enthaltenen **MED**-Musikeditor erstellt. Das Musikprogramm setzt sich aus einem bedienungsfreundlichen Editor und einer Abspielroutine zusammen, mit der man die erstellten Musikstücke abspielen kann, ohne auf den Editor zugreifen zu müssen.

Mit Public Domain zum Heim-Musiker

In dem Editor werden die Notenwerte und die Instrumente eingegeben, die dann letztendlich den fertigen Sound ergeben. Der Player liest die Werte ein und spielt sie dann ab. Im Player können einzelne Passagen des Musikstückes analysiert und überarbeitet werden. Dazu dient die Blockgestaltung des Players.

Das Musikstück wird in einzelne Blöcke unterteilt, wobei auch die Möglichkeit besteht, einzelne Blöcke abzuspielen, zu verändern und umzustellen.

Das Programm **NECquick** ermöglicht die problemlose Einstellung gewünschter Parameter an NEC-Druckern.

NEC als Schriftsatzschönling

Zwar lassen sich viele Druckeroptionen wie NLQ, Pica und Elite manuell über die Fonttaste am Drucker auswählen, jedoch lassen sich damit nicht die Styles (Bold, Italic, etc.) definieren. Die Einstellungsmöglichkeiten präsentieren sich in einem übersichtlichen Menü. Der NEC-Drucker bietet seinem Besitzer 256 Schriftkombinationen, von denen eine Schriftart gezielt angewählt werden kann. NECquick besitzt zudem Optionen für Fett-, Kursiv- und Schattendruck, sowie Unterstreichen, die Einstellung des Zeichensatzes, vertikale und horizontale Vergrößerung, den Zeilenvorschub und Zeilenabstand.

Disketten unter der Lupe

Ein Diskettenmonitor listet alle auf der Diskette befindlichen Daten auf. Durch die Sektor- und die Blocknummer werden Bereiche der Diskette genauestens analysiert. Das Programm **DiskBuzz** nimmt in der Sparte der Diskettenmonitore eine führende Rolle ein. Mit einem gut durchdachten Menü und einer Vielzahl an Optionen gestaltet sich DiskBuzz zu einem Spitzenmonitor. Der Monitor prüft alle Sektoren der Diskette auf die korrekte Checksumme und gibt an, wenn ein Fehler vorliegt. Weiterhin kann man nach einem frei wählbaren String suchen oder die belegten Blöcke anzeigen und auf einem Drucker ausgeben lassen.

Das nächste Programm ist eine Demoversion des Relo-Diskmasters. Hiermit lassen sich Diskettenoperationen vereinfacht durchführen. So ermöglicht das Programm neben der übersichtlichen Darstellung von Disketteninhalten das Erstellen von Unterdirectories oder Schützen von Dateien mittels Protect-Befehl. Aber auch Umbenennen und das Kopieren von Files fließen in das Programm ein. Dabei werden Uhrzeit und der freie Speicherbereich in einer Kopfzeile angezeigt.

(Jürgen Seibel/br)

Name: Time 5
Quelle: APS Electronic

Ein Bereich der sowohl im PD-Sektor als auch durch kommerzielle Programme vernachlässigt scheint, ist der der Mathematikprogramme. Das Programm **R.O.M.** tut seinen Dienst zur Sache, um die entstandene Lücke etwas zu schließen. Es besteht zum Hauptteil aus zwei Komponenten: einem wissenschaftlichen Taschenrechner und einem Programmteil zur grafischen Darstellung von Funktionen.

Mathe-As statt Mathe-Haß

Nach dem Einladen präsentiert sich ein Hauptmenü, das sich in drei Bereiche gliedert. So ist ein Bereich für die Variablendeklaration zuständig. Hier lassen sich die Variablen von A bis Z mit Werten definieren. Der nächste Menüpunkt beinhaltet einen Taschenrechner. Mit diesem Rechner lassen sich trigonometrische Funktionen oder Algorithmen errechnen. Nun gelangen wir zum dritten und wohl interessantesten Punkt.

Er beschäftigt sich mit der Verarbeitung von Funktionen. Nach der Eingabe einer Funktion wird sie in den wählbaren Definitionsbereich auf einem Koordinatensystem grafisch ausgegeben. Der so entstandene Graph läßt sich zusätzlich ausdrucken. Mit einer Zeichen-Option ist es dem Anwender möglich, einen Graphen selbst zu zeichnen. Durch die Werte kann man dann Rückschlüsse auf die Funktion ziehen.

(Jürgen Seibel/br)

Name: Oase 20
Quelle: Wolf Computertechnik

Eine kleine Übersicht der Vertreiber von Public Domain, Free- und Shareware sowie Prüf-vor-Kauf-Programmen (ohne Anspruch auf Vollständigkeit)

A.P.S. Electronic
Sonnenborstel 31
3071 Steimbke
Tel.: 05026/1700

BAVARIANSOFT
Postfach 72
8473 Pfreimd
Tel.: 09606/286

BELLINGRATH, CHRISTIAN
Hans-Böckler-Str. 55
5860 Iserlohn
Tel.: 02371/24192

DIGITAL MARKETING D. MÜCKTER
Krefelder Str. 16
5142 Hückelhoven-Baal
Tel.: 02435/2086,428, oder 1295

HIESKE, DIETER
Schillerstr. 36
6700 Ludwigshafen
Tel.: 0621/673105

KEIM, PETER
Vogelsanger Str. 34
5000 Köln 30
Tel.: 0221/520765

MAXON COMPUTER GMBH
Industriestr. 26
6236 Eschborn
Tel.: 06196/481811

Ossowski, Stefan
Veronikastr. 33
4300 Essen 1
Tel.: 0201/788778

TechnikSupport Verlag GmbH
Bundesallee 36-37
1000 Berlin 31
Tel.: 030/8621314

WOLF Computertechnik
Deipe Stegge 187
4420 Coesfeld
Tel.: 02541/2874

Österreich

KÜPPERS, BERND
Felberstr. 7
A-5730 Mittersill
Tel.: 06562/282

PETER RAUSCHERS COMPUTERSHOP
A-1100 Wien
Weldengasse 41
Tel.: 0222/621535

Public-Domain-Spieleshow

Sicher, viele kommerzielle Spiele sind die 50 bis 80 Mark, die sie kosten, auch wirklich wert. Aber leider kommt es nur zu oft vor, daß die teuren Spiele enttäuschen. Mit Public Domain kann Ihnen das nicht so schnell passieren. Hier kostet eine ganze Disk voller Spiele um die fünf Mark. Und sollte man sich tatsächlich völlig verkauft haben, und der blanke Unsinn befindet sich auf dieser Disk, formatieren Sie sie, dann haben Sie für fünf Mark eben eine Leerdisk erstanden.

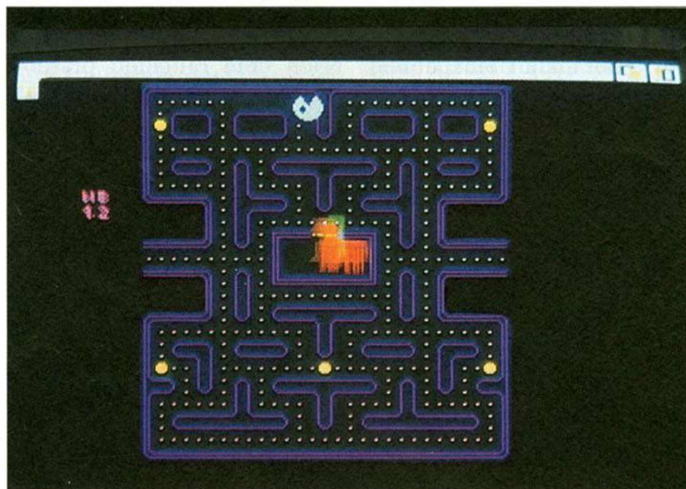
Maze Man

Der Urvater aller Labyrinthspiele ist zweifelsohne der allseits beliebte Pac Man. Dieses Spiel gibt es schon in verschiedenen Public-Domain-Versionen, von denen Maze Man eine der überzeugendsten ist. Maze Man ist ein kugeliger, blauer Geselle, der in einem Labyrinth haust und sich von den darin verteilten Pillen ernährt. Aber der kleine Pillenfresser lebt nicht alleine im Labyrinth. Er muß es mit drei Widersachern teilen, die nichts anderes im Sinn haben, als ihn zu jagen. Nur geschicktes Manövrieren kann den blauen Helden retten. Damit er nicht ganz wehrlos dasteht, gibt es in jedem Labyrinth einige Kraftpillen, die aus dem friedlichen Maze Man ein echtes Monster machen, das sogar seine Widersacher auffressen kann. Jedesmal wenn Maze Man in einem Labyrinth alle Pillen zu sich genommen hat, folgt eine Geschicklichkeitsrunde. Dabei sind nur die Pillen zu sehen, nicht aber die Wände des Labyrinthes. Da kann es schnell passieren daß Maze Man, während ihm ein Monster auf den Fersen ist, in eine Sackgasse läuft. Ist die Geschicklichkeitsrunde geschafft, geht es in einem neuen Labyrinth weiter, das ein wenig anders aussieht als das erste. Maze Man ist tadellos programmiert, wird von lustigen Soundeffekten untermalt und präsentiert sich als durch und durch spielbares PD-Prachtstück. Echt empfehlenswert.

Name: Maze Man
Enthalten auf: Cactus 15
Vertrieb: Siehe Anbieterliste

Return to Earth

Menschen und Roboter waren von der Erde in die Tiefen des Weltraums aufgebrochen und gründeten ein galaktisches Imperium, das Millionen von Sternensystemen umfaßte. Doch Neid, Mißgunst und Zwietracht ließen es binnen kurzer Zeit zerbrechen. Kurz vor dem Ruin schloß sich jedoch eine Gruppe von oppositionellen Wissenschaftlern zusammen und wurde gezwungen, ins Exil zu gehen. Sie wählten sich als Zufluchtsort den am äußersten Rand der Galaxie gelegenen Planeten Terminus, um von dort die letzten Reste des alten Imperiums zu einem neuen, dauerhaften Staatenverbund zu formen. Die nachfolgenden Generationen brauchten für dieses Vorhaben etwa 1000 Jahre, so wie es der Gründer von Terminus vorausgesagt hatte. Während dieser Zeit verstreuten sich die Menschen über viele Galaxien hinweg, und die Erinnerung an den Mutterplaneten Erde verblaßte zusehends. Jedoch die Begegnungen mit anderen Lebensformen und Zivilisationen ließ bei vielen Menschen die Frage nach der eigenen Herkunft erneut aufkommen.



Labyrinthfreunde aufgepaßt, Maze Man entführt Euch in einen Irrgarten, der es in sich hat



In ferner Zukunft ist die Erde Legende. Werden Sie es schaffen, sie wiederzufinden?

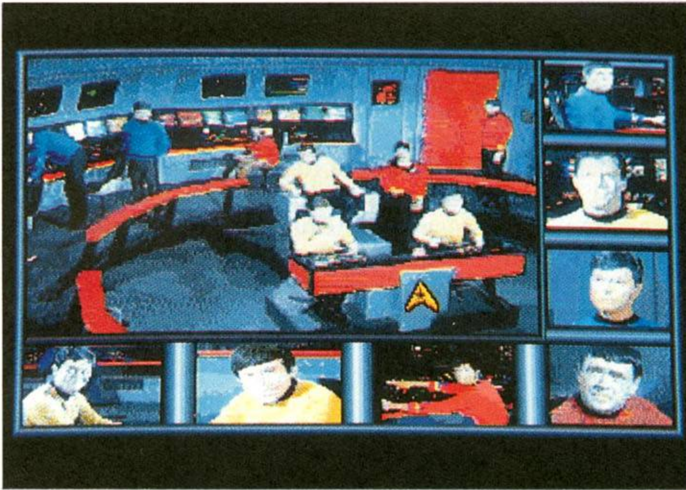
Nun treten Sie in Aktion. Als Golan Trevize, Ratsherr von Terminus, stellen Sie sich 500 Jahre später dieselbe Frage und brechen auf, die Wiege der Menschheit zu finden – die Erde. Zu diesem Zweck rüsten Sie Ihr Raumschiff aus und verlassen Terminus mit unbekanntem Ziel....

Dies ist Ihre Mission, jedoch ist es nicht nur damit getan, einfach durchs Weltall zu düsen, um die gute Mutter Erde zu suchen. Viele Gefahren warten auf Sie, wie z.B. feindliche Raumschiffe, die nur darauf aus sind, Sie und Ihre Besatzung ins Nirwana zu befördern. Jedoch nicht nur lichtscheues Gesindel treibt sich im All herum, nein, manche Raumschiffe können Ihnen auch hilfreiche Auskünfte über Ihre Mission geben.

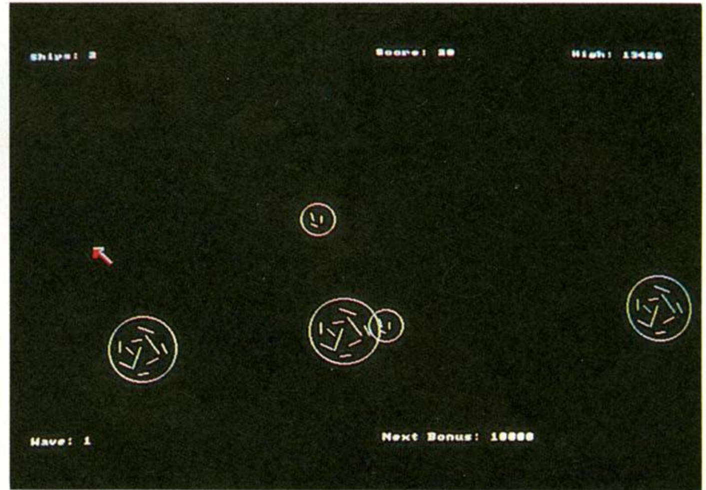
Damit Sie sich Ihre Brötchen für unterwegs verdienen können, ist es Ihre wichtigste Aufgabe, die einzelnen Planeten anzufliegen und auf den die verschiedenen Planeten umfliegenden Raumstationen Handel zu betreiben. Auf diesen Raumstationen können Sie auch eventuelle Schäden an Ihrem Raumschiff ausbessern lassen. Es kann durchaus passieren, daß Ihnen auf manchen Stationen ein Sondertransportauftrag angeboten wird.

Sobald Sie eine Raumstation angeflogen haben, öffnet sich ein neuer Bildschirm, bei dem Sie all Ihre Ein- und Verkäufe tätigen können. Haben Sie sich gut genug versorgt, können Sie Ihre Suche fortsetzen. Nachdem Sie nun die Raumstation verlassen haben, müssen Sie natürlich Ihren Zielort angeben. Dies geschieht mittels der Sternenkarte, bei der Sie sich über die einzelnen Planeten und deren Population erkundigen können.

Ihr Raumschiff ist sehr gut ausgerüstet und verfügt unter anderem über ein Gerät, mit dem herannahende Raumschiffe identifiziert werden können. Somit können Sie rechtzeitig entdecken, ob es sich um freundlich gesonnene Besatzungsmitglieder handelt, mit denen Sie, wie bereits erwähnt, Handel treiben können, oder ob Sie die Laser



Auch ohne spitze Ohren in die Rolle von Mr. Spock schlüpfen? Public Domain Star Trek macht's möglich



Chaos im Asteroidenfeld, da darf man den Überblick nicht verlieren

(bzw. Raketen) schon einmal vorglühen lassen müssen.
Return to Earth ist eine Handelssimulation, vermischt mit einigen Actioneinlagen, die wieder einmal beweist, daß Low-Cost-Produkte nicht so einfach abzuweisen sind. Ich kann jedem Elite-Freund dieses Spiel wärmstens empfehlen, da nicht nur die Grafik, sondern auch der Sound sehr gut gelungen ist. Es muß jedoch dazu gesagt werden, daß zwar das Programm an sich unter den Titel Shareware fällt, jedoch die achtseitige Anleitung ohne Genehmigung nicht kopiert werden darf.

Name: Return to Earth
Besonderheiten: Shareware
Quelle: Rainer Wolf Computer Service

Star Trek

Noch immer ist James T. Kirk mit seiner Besatzung in den unendlichen Weiten des Weltalls auf der Suche nach Geheimnissen, die für die Menschheit interessant sein könnten. Zwischen zwei Begegnungen der dritten Art bekommt er den Auftrag, eine Anzahl von Geheimpapieren mit besonders wichtigen Informationen wiederzubeschaffen. Die Sicherheit der Föderation hängt davon ab. Das wissen natürlich auch die feindlichen Klingonen, die ihrerseits das Halali blasen, um in den Besitz der wichtigen Papiere zu gelangen.
In Star Trek steuert der Spieler die gesamte Crew der Enterprise. Jedes Besatzungsmitglied hat seinen Aufgabenbereich. So ist Spock für die Auswertung der Planetendaten verantwortlich, Uhura erledigt den Funkverkehr, Checkov schlägt sich mit den Klingonen herum und so weiter.

Auf der Suche nach den Geheimpapieren werden nun die einzelnen Systeme angeflogen. Eine Planetenkarte gibt Auskunft über die Anzahl der Planeten pro System. Mr. Zulu bringt uns auf Wunsch in die Umlaufbahn eines dieser Himmelskörper. Es kann nun eine Analyse desselben durchgeführt werden, die uns über seine Beschaffenheit aufgeklärt.

Auf den Flügen kommt es häufig zu Begegnungen mit den Klingonen, die sich nur durch massives Feuern dazu bewegen lassen, die eigene Mission nicht weiter zu stören.

Herausragend an Star Trek ist die Präsentation. Alle Grafiken sind voll digitalisierte Aufnahmen aus der Serie. Auch der Sound wurde direkt übernommen, so daß sich alle diejenigen, die sich ein wenig mit der Serie oder den Filmen auskennen, gleich heimisch fühlen werden.

Die Steuerung erfolgt bei Star Trek komplett per Maus. Aus dem Cockpit heraus werden die einzelnen Mitglieder der Crew angeklickt. Man gelangt so in Untermenüs, in denen dann die einzelnen Aktionen getätigt werden. Leider wird das Anklicken vom Programm weder durch einen Ton noch durch eine grafische Veränderung bestätigt, so daß man stets einige Momente warten muß, um zu wissen, ob das Programm den Befehl verstanden hat bzw. ausführen kann.

Auch hätte dem Programm eine Art Statusanzeige gutgetan. Man muß sich schon merken, wie schnell man gerade fliegt oder welche Waffensysteme aktiv sind.

Die Aktionsmöglichkeiten sind, betrachtet man den Aufwand, mit der Grafik und Sound erstellt worden sind, etwas dürftig. Das führt dazu, daß das Spielgeschehen auf die Dauer ein wenig monoton wird.

Meine Meinung: Trotz Mängel empfehlenswert. Wenn man solch ein professionell gestaltetes Programm für ein paar Mark zu kaufen bekommt, dann wäre es nicht fair, die wenigen Mängel zu stark zu bewerten; denn für ein Public-Domain-Programm ist Star Trek eigentlich zu gut. Es muß keinen Vergleich mit anderen Programmen scheuen.

Name: Star Trek
Enthalten auf: Oase 21
Besonderheiten: 1 MByte Speicher notwendig
Vertrieb: Siehe Anbieterliste



Die Invasion der dreibeinigen Kampfroboter muß abgewehrt werden

Asteroids

Ein winziges Raumschiff kämpft sich durch einen wilden Meteoritenhagel. Die wild wirbelnden Felsbrocken drohen das kleine Schiff zu zerschmettern. Die einzigen Mittel, die dem Schiff zu Gebote stehen, um eine drohende Havarie zu vermeiden, sind ein kleiner Laser und der Antrieb. Mit dem Antrieb kann den umhertreibenden Felsen ausgewichen und in eine günstige Position ma-

növriert werden, um den Asteroiden auf den Leib zu rücken. Wenn allerdings einer der Felsen getroffen wird, zerspringt dieser in kleinere Fragmente, die ihrerseits umhertreiben. Wird wild in der Gegend herumgeballert, ist der Bildschirm mit vielen kleinen Felsen übersät und entsprechend gefährlich.

Basierend auf einem uralten Spielhallenhit präsentiert sich Asteroids als eine wirklich überzeugende PD-Version dieses Programms. Es ist zwar komplett in schwarzweiß gehalten, und das Bild flickert durch den gewählten Bildschirmmodus, ist aber trotzdem sehr spielbar geworden.

Name: Asteroids
Enthalten auf: Kick A63
Vertrieb: Siehe Anbieterliste

Triclops Invasion

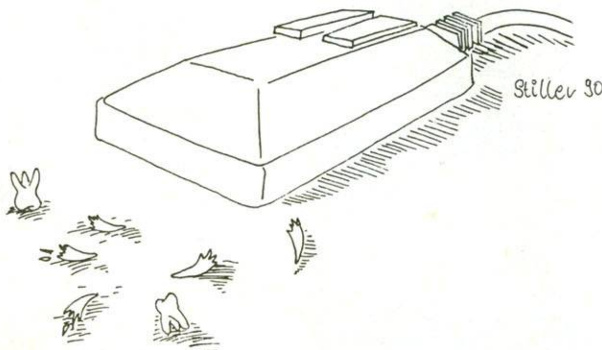
H. G. Wells hat sie sich ausgedacht, und heute bedient sich ihrer alle Welt, ohne genau zu wissen, woher sie überhaupt stammen. Die Rede ist von den dreibeinigen Herrschern oder Triclops, wie sie auch genannt werden. Diese Kampfmaschinen, denn um nichts anderes handelt es sich bei einem Triclop, wurden von einer Rasse fieser Außerirdischer als Invasionswerkzeug entwickelt. Triclops Invasion versetzt Sie ohne lange Umschweife in die Rolle des Verteidigers eines Forts auf unserer guten, alten Erde. Allerdings sind die dreibeinigen Invasionsgeräte bereits gelandet und befinden sich im Anmarsch auf das Fort. Nun gilt es, mit einem Jet und zwei Panzern den Angriff abzuwehren.

Triclops ist eines der professionellsten Public-Domain-Spiele auf dem Markt. Obwohl der Programmablauf mitunter ein wenig zäh ist und so einiges an Wartezeiten anfällt, vermag Triclops zu überzeugen. Die Welt, die es zu verteidigen gilt, wird in einer tollen Fraktalgrafik dargestellt, die von Spiel zu Spiel neu generiert wird.

Natürlich kann der Spieler zwischen seinen drei Verteidigungsvehikeln hin und her schalten, aber er ist darüber hinaus in der Lage, die Kameraposition zu bestimmen. So kann man sich entweder den Schauplatz des Getümmels aus der Vogelperspektive oder aber aus dem Blickwinkel eines seiner Fahrzeuge ansehen.

(br/jf/hs)

Name: Triclops Invasion
Enthalten auf: RW2
Vertrieb: Siehe Anbieterliste



Softwarenoten und wie sie zustandekommen

Daß Ihre AMIGA DOS bei den Spiele-Tests ein Urteil vergibt, haben Sie selbst sicherlich schon gesehen. Wie dieses Urteil funktioniert, das soll hier noch einmal im Detail erklärt werden.

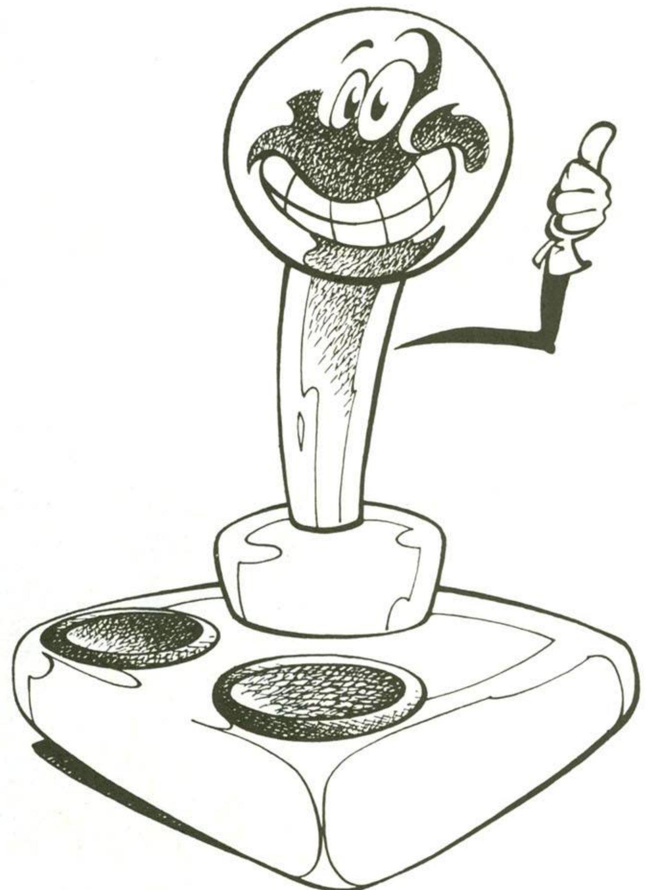
AMIGA DOS beurteilt jedes Spiel anhand verschiedener Kriterien: Die Grafik, die Musik und den Spielspaß. Die Noten dafür werden in der kleinen Urteilsbox, die am Ende jeder Review zu finden ist, unter den Oberbegriffen Grafik, Sound und Motivation dargestellt. Die Note selbst richtet sich nach der Höhe des wiedergegebenen Wertes. Je höher, desto besser. Zu diesen Noten gesellt sich ferner ein individuelles Urteil des Testers. Eine vierte Note also, die allerdings nicht wiedergegeben wird.

Aus diesen vier Werten bildet sich schließlich das Gesamturteil:

- **Sehr gut**
- **gut**
- **o.k.**
- **geht so**
- **schlecht**



Die Bedeutung dieser Bewertungen erklärt sich von selbst. Uns bleibt noch, Ihnen viel Spaß mit dem AMIGA DOS Spieleteil zu wünschen.





LEISURE SUIT LARRY

IS LOOKING FOR LOVE

Larry ist ein Mann von Welt. Er kann es sich leisten, Gesicht zu zeigen, um genau zu sein, bis zur Fontanelle. Auch wenn sein Geldbeutel nicht so dick ist, wie er sich das wünschen würde, ist er ein begehrenswerter Mann, meint er.



Hollywood, Großstadt und Medienmetropole und die Heimat von Larry Laffer. Larry hat den Ernst des Lebens erkannt. Eine Erfahrung, die zur Folge hat, daß er versucht, so gut es geht, den Problemen des täglichen Alltags aus dem Weg zu gehen. Aber mitunter läßt sich dies nicht vermeiden. Eve, sein bisheriger Lebensanker, hat Larry schnöde die Tür gewiesen. Der einzige Lichtblick an diesem üblen Tag ist der Dollar, den Larry in der Garage findet. Zurück nach Hause kann er nicht, also stellt sich Larry der Herausforderung, mit einem Dollar in der Tasche eine tolle Zeit zu haben.

Durch geschicktes Agieren in den KROD TV Studios gelingt es Larry, nicht nur eine Kreuzfahrt zu gewinnen, sondern auch noch den Superbowl bei der TV-Lotterie, Jahr für Jahr eine Million Dollars cash auf die Hand. Das dabei nicht alles mit rechten Dingen zugeht, muß auch die Frau feststellen, die zusammen mit Larry die Kreuzfahrt gewann, denn Larry ist genau der Typ, mit dem sie so wenig wie möglich zu tun haben möchte.

Eine Kreuzfahrt, dazu gehören Badehosen und Sonnencreme, beides Dinge, die Larry sich noch besorgen muß. Als millionenschwerer Schickimicki muß es natürlich etwas Exklusiveres sein, trotzdem sind die Preise der Molto Lira Boutique eine echte Erfahrung. 180.000 $ sind schon ein kerniger Preis für eine Badehose. Selbst für den Mann von Welt.

Aber, nachdem noch einige weitere Besorgungen gemacht wurden, hat Larry sich soweit ausgerüstet, daß die Kreuzfahrt endlich losgehen kann. Seinen Ausweis hat er in letzter Minute aus Eves Mülltonne retten können, die Beziehung scheint er ja endgültig vergessen zu können.

Auf der U.S.S. Love will Larry sich richtig entspannen und sich ein wenig der holden Schönen widmen, die zusammen mit ihm die Reise gewann. Jedoch hat diese sich verkrümelt und statt ihrer nimmt die monströse Mutter teil, die teilweise wirklich extrem veranlagt ist. Zwischenzeitlich ist es Larry aber auch gelungen, in einer KGB-Fahndungsliste aufgenommen zu werden, was auch für die Zukunft noch turbulente Abenteuer garantieren dürfte. Verwechslung ist angesagt, denn Larry ähnelt dem Kontaktmann des geheimnisvollen Dr. Nonookee aufs Haar und bekommt deshalb, ohne es zu wollen, ein Onklunk (Musikinstrument). Als der richtige Kontaktmann auftaucht, hat Larry auch diesen auf den Fersen. Für diese heitere Hatz haben sich die Macher bei Sierra wieder Feines ausgedacht: Über Luxusdampfer, einsame Inseln, Vul-



Larrys Abenteuer führen ihn und Sie an exotische Schauplätze



Der freundliche Nachbar vom KGB hat ein paar Fragen, auf die er Antwort sucht

kan Atolle, dampfenden Dschungeln oder einfach dem Dickicht diverser Städte, reicht das Sortiment der Schauplätze, die bereist werden.

Schlüpfen Sie in die Rolle von Larry Laffer und erleben Sie ein paar haarsträubende Abenteuer. Darum geht es im wesentlichen. Sie steuern und kontrollieren den Helden, Sie bestimmen, was er tun soll (allerdings bestimmt nach wie vor das Programm, was gemacht werden kann). Die Ereignisse, die Sie erwarten, haben einen eindeutig zweideutigen Charakter und der Humor einen deutlich schlüpfrigen Unterton, ohne dabei geschmacklos zu werden. Trotz vieler Grafiken gehört auch zu diesem Spiel eine gehörige Portion erläuternder Text, den das Spiel in bestimmten Situationen ausgibt. Im Gegenzug muß der Spieler, sobald er Larry speziellere Aktionen ausführen lassen will, dies dem Spiel über eine Texteingabe mitteilen. Leider sind alle Texte in Englisch. Diese sind jedoch nicht so komplex, so daß sie sich unter Zuhilfenahme eines Wörterbuches ohne große Probleme übersetzen ließen. Trotzdem, wer gar kein Englisch kann, der sollte von diesem Spiel Abstand nehmen. Was bleibt, ist ein Abenteuerspiel, mit einer sehr originellen Story, die auch Erwachsenen (oder gerade ihnen) viel Spaß bringen kann. Larrys erotische, chaotische Abenteuer sind für mehr als einen lauten Lacher gut.

(hs)



AMIGA DOS Blitzlicht

Name: Leisure Suit Larry goes Looking For Love
Hersteller: Sierra
Quelle: Fachhandel
Preis: 119,95 DM

Positiv:
- originelle Handlung
- gute Musik
- passable Grafik

Negativ:
- unterstützt keine Zweitlaufwerke





Flippern in der dritten Dimension – verbunden mit der Vogelperspektive entsteht daraus ein heilloses Durcheinander

1st Person Pinball

Flippersimulationen sind seit jeher ein beliebtes Genre in Heimcomputerkreisen. Auf die Idee, ein Spiel in 3D-Perspektive aus der Sicht des Balles, kombinierbar mit einem Überblick über das Spielfeld, zu programmieren, ist bis jetzt noch niemand gekommen...

Das Programmierteam von Tyne-Soft hat sich dieser Aufgabe angenommen. Das Ergebnis dieser Mühen liegt uns in Form des Spiels 1st Person Pinball vor. Ein Flipper, der nicht nur den bei solchen Spielen gewohnten Anblick beinhaltet, sondern auch einen 3D-Ausblick aus der Sicht der Kugel. Die einzelnen Darstellungsweisen können frei gewählt werden. Spielt man in der reinen 3D-Perspektive, kann sehr leicht die Orientierung verlorengehen, obwohl der Blickwinkel, aus dessen Sicht das Spielgeschehen dargestellt wird, vom Spieler vor oder hinter der Kugel frei positioniert werden kann. Der zu sehende Ausschnitt ist jedoch ziemlich klein und die Bewegungen der Kugel sehr schnell, was die Orientierung ungemein erschwert.

Ist das Spiel so eingestellt, daß sowohl der 3D-Ausblick als auch der normale Flipper zu sehen sind, wird der Programmablauf deutlich langsamer, was dem Spielspaß nicht gerade zuträglich ist.

An sich ist 1st Person Pinball mit reinen 3D-Grafiken kaum spielbar. Schaltet man die 3D-Grafik aus, um einen normalen Flipper zu sehen, gelangt man bald zu dem Schuß, daß es hier kaum etwas gibt, was man nicht irgendwo anders schon einmal besser gesehen hat. Und für einen reinen 3D-Gag ist 1st Person Pinball ein bißchen zu teuer.

(Robert Marz/hs)

AMIGA DOS Blitzlicht

Name: 1st Person Pinball
Hersteller: Tynesoft
Quelle: Fachhandel
Preis: 64,95 DM

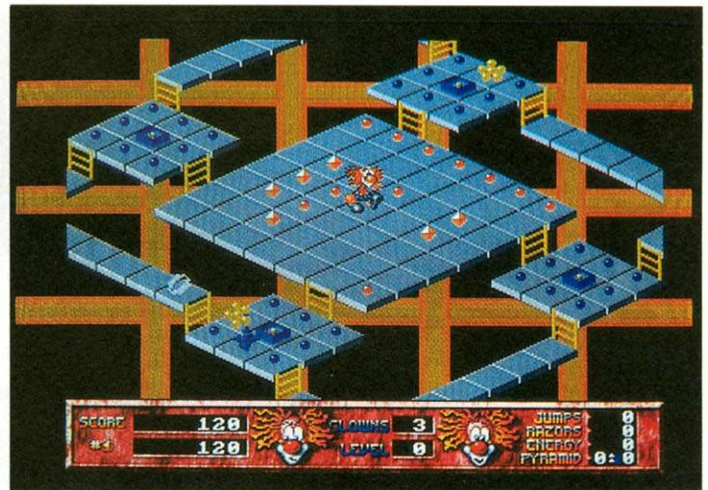
Positiv:
- guter Sound
- Parameter modifizierbar
- bis zu vier Mitspieler

Negativ:
- 3D-View etwas wackelig und verwirrend
- unzulängliche Tastaturabfrage bei Eintragungen in die Highscoreliste





Gerade der Wind spielt eine große Rolle beim Abschlag – Finden Sie die richtige Linie



Clown Beppo auf Juwelenhatz im Kristalland

Jack Nicklaus

Greatest 18 Holes of Major Championship Golf

Wie in jeder Sportart gibt es auch im Golf einen gefeierten Alleskönner. Der Tausendsassa Jack Nicklaus gewann jedes im Golf relevante Turnier und dominierte 30 Jahre lang das Golfgeschehen.

Daß Nicklaus etwas vom Golf versteht, verdeutlicht schon die Auswahl der Plätze. Mit Desert Mountain GC und Castle Pines GC präsentieren sich zwei von ihm gestaltete Golfplätze, die es in sich haben. Tatsächlich beinhaltet Jack's Greatest 18... die 18 Löcher verschiedener Golfplätze, die Mr. Nicklaus für die besten der Welt hält.

Um spielerische Akzente zu setzen, enthält Jack's Greatest... zwei Wettkampfmodi: Skins-Play und Stroke-Play.

Das Ziel des Skins-Play ist der Gewinn eines Loches. Dabei sind die einzelnen Löcher mit verschiedenen Geldbeträgen dotiert, so daß Sie mittels der Löcher Geld erwerben können. Beim Stroke-Play kann sich der Spieler mit der niedrigsten Schlagzahl zum Sieger küren lassen. Ein Fenster eröffnet Ihnen den Blick zum Loch aus der Vogelperspektive. So können geografische Besonderheiten genau analysiert und der richtige Schläger gewählt werden.

Beeindruckend ist die grafische Umsetzung der Spielidee, die zusammen mit dem technisch einwandfreien Programm und der durchdachten Menügestaltung für ein echtes Spielerlebnis gut ist.

(Jürgen Seibel/hs)

AMIGA DOS Blitzlicht

Name: Jack Nicklaus' Greatest 18 Holes of Championship Golf
Hersteller: Accolade
Quelle: Fachhandel
Preis: 84,95 DM

Positiv:
- Auswahl der Golfplätze und Löcher
- naturgetreue Umsetzung
- kein Kopierschutz
- unterstützt zweites Laufwerk

Negativ:
- unterstützt bei 512 kByte nur ein Laufwerk



Clown-O-Mania

Beppo, der Clown, hat sein Leben lang das Publikum zum Lachen gebracht. Jetzt hat er eine Karte gefunden, auf der der Weg zum Land der Kristalle verzeichnet ist.

Mit all den Juwelen könnte Beppo sicherlich ein sorgenfreies Leben führen, deshalb bricht er auch gleich auf. Jedoch ist das Kristalland ein sonderbarer Ort, der auch seine Gefahren birgt.

Wer Beppo begleitet wird mit einer Welt konfrontiert, die aus aneinandergereihten Quadern besteht, die inmitten eines bodenlosen Abgrundes schweben. Die Quader bilden wild verwinkelte Wege, türmen sich in mehreren Stockwerken übereinander und sind mitunter durch Leitern untereinander verbunden. Zwar schützen unsichtbare Wände Beppo vor dem Abstürzen, aber dafür geistern auf den Plattformen allerlei Ungeheuer umher, die Beppo ans Clownleben wollen. Aber auch hier gibt es Schutz in Form von Rasierklingen, die Beppo sammeln kann. Werden sie im geeigneten Moment auf die Gegner geschleudert, verschwinden diese für immer.

Durch 70 verschiedene Zonen, deren Schwierigkeitsgrad kontinuierlich ansteigt, muß Beppo seinen Weg finden, der nicht immer leicht ist. Da kommt es auf Timing genauso an, wie auf schnelle Reaktionen und einen guten Joystick.

Obwohl Clown-O-Mania grafisch nicht gerade der letzte Schrei ist, vielmehr sogar irgendwo im Mittelfeld anzusiedeln ist, spielt es sich erstaunlich gut. Daran sind nicht zuletzt die gute Musik und die netten Extras im Spiel schuld. Denn gerade diese vermitteln dem, ansonsten eher eintönigen Programm den Hauch Strategie, der es spielbar macht.

(hs)

AMIGA DOS Blitzlicht

Name: Clown-O-Mania
Hersteller: Starbyte
Quelle: Fachhandel
Preis: 64,95 DM

Positiv:
- Strategie und Geschicklickeit gefordert
- hörenswerter Sound

Negativ:
- hausbackene Grafik





Das Königreich von Albareth ist der Schauplatz einer abenteuerlichen Suche

Times of Lore

Seit 700 Jahren leben Menschen und Elden unter der Herrschaft des hohen Königs in Frieden und Eintracht nebeneinander. Drei magische Kleinode wurden geschaffen, um den Bund der Menschen und Elden zu festigen: die Tafel der Wahrheit, das Medaillon der Kraft und magische Steine, die in der Lage sind, die Zukunft vorherzusagen.

Die Artefakte wurden in die Hände treuer Lords gelegt, und Frieden herrschte im ganzen Land. Dieser endete jedoch abrupt mit der Ankunft der Barbaren, die von Süden her angriffen... Zehn Jahre tobte der Kampf, bis der König selbst beschloß, mit seinen besten Rittern gen Süden zu ziehen und die Barbaren Mores zu lehren. So wogte der Kampf, aber keine Seite konnte den Sieg erringen. Schließlich benutzte der König die Macht des Medaillons, um die Schlacht anzuhalten und mit Heidric, dem Anführer der Barbaren, zu sprechen. Viele Tage redeten sie miteinander und gelangten schließlich zu einer Einigung. Die südlichen Länder wurden den Barbaren zu eigen gegeben, doch mußten sie dem hohen König Treue schwören.

Nun war Valwyn von den Mühen des zehnjährigen Krieges so erschöpft, daß er sein Kind nahm, um in die Heimatländer der Elden zurückzukehren und seine Kräfte zu erneuern. Das Königreich ließ er in der Obhut eines treuen Verwalters und versprach, nach zwanzig Jahren zurückzukehren. Das war das letzte, was je von ihm vernommen wurde.

Dariel, der eingesetzte Reichsverweser, war nun nicht in der Lage, Frieden und Wohlstand aufrechtzuerhalten. Viele Eldenlords hegten immer noch Haß gegen die Barbaren, und schließlich versuchten sie, die südlichen Länder von den Barbaren zurückzuerobern.

So gab es wieder Krieg im Lande. Als auch noch Ungeheuer vom Norden her einfielen, stellte man fest, daß das Land einen Retter braucht. Sie!

Sie können wählen, welchen Charakter Sie darstellen wollen. Ein Ritter ohne Furcht und Tadel, eine Walküre sowie ein Barbar stehen zur Wahl, um dem Land den Frieden zurückzubringen. Eine lange und gefahrvolle Reise muß nun angetreten werden, um die magischen Artefakte wiederzubeschaffen. Dies ist allerdings nicht gerade einfach. Außerhalb der Ansiedlungen heulen die Monster ihr grausiges Lied, und auch die Wanderer, die man unterwegs treffen kann, sind mit Vorsicht zu genießen. Im Zweifelsfall empfiehlt es sich, erst zu schlagen und dann zu fragen.

Viele der Monster, Wegelagerer und sonstigen Zeitgenossen führen übrigens wichtige Gegenstände mit sich. Tränke, Schriftrollen, Waffen und Geld sind nur einige der Dinge, die man nach einem erfolgreichen Kampf aufsammeln kann. Die schwindenden Lebenskräfte lassen sich mit geeigneten Tränken auffrischen. Den gleichen Zweck erfüllt auch eine Übernachtung in einer Herberge, wo außerdem der aktuelle Spielstand gespeichert wird.

Hier kann man auch Rationen kaufen, um für längere Wanderungen gerüstet zu sein.

Wichtig ist auch das Gespräch. Nur so können Sie sämtliche Informationen zusammentragen, die zur Lösung der Quest notwendig sind. So, nun liegt es nur noch an Ihnen, wann Frieden und Wohlstand wieder Einzug halten. Frisch ans Werk!

Times of Lore präsentiert sich als Abenteuer, das grafisch leider nur als zweckmäßig einzustufen ist. Komplett menügesteuert ist die Handhabung der Spielfigur unproblematisch. Mit Hilfe der Spacetaste wird von der Steuerung der Figur auf die Statuszeile umgeschaltet, deren Optionen nun aufgerufen werden können. Eine Pausenfunktion läßt auch während des Spieles die Nahrungsaufnahme zu. Trotz der spärlichen Grafik macht Times of Lore Spaß und lädt, solange die Kernaufgabe noch nicht gemeistert ist, immer wieder zum Spielen ein.

(mm/hs)



Drei magische Schätze gilt es zu finden, um den Frieden zu wahren

AMIGA DOS
Blitzlicht

Name: Times of Lore

Hersteller: Origin

Quelle: Fachhandel

Preis: 89,– DM

Positiv:
- komplett menügesteuert
- langer Spielspaß

Negativ:
- magere Grafik
- langsamer Spielablauf

AMIGA DOS URTEIL OK

GRAFIK	SOUND	MOTIVATION
5	4	8



Auf dem Weg zum Drachen Meka gibt es einiges zu erleben

Wonderboy in Monsterland

Im Wunderland ist der Teufel, äh, der Drache los. Meka, ein feuerspeiendes Ungeheuer, hat die Macht an sich gerissen, da muß Wonderboy ran.

Es gibt Spiele, die kommen grimmig, blutrünstig einher. Dann gibt es Spiele, die sich durch und durch ernstnehmen. Und glücklicherweise gibt es da noch die Spiele, die Spaß machen, voller Ideen stecken und lustig sind. Eigenschaften, die Wonderboy voll und ganz in sich vereint. Der zu Beginn des Spiels in Windeln gekleidete Held muß die vielen Landschaften des Wunderlandes durchqueren, um zur Behausung des Drachen zu gelangen. Dieser Weg wird aber bereits von den Helfershelfern des Untiers bewacht, so daß Wonderboy sich seinen Weg mit einem Schwert erkämpfen muß. Besiegte Gegner hinterlassen Geldstücke oder Juwelen, die Punkte bringen. Unterwegs kommt er dabei durch Städte und an anderen Behausungen vorbei, die er betreten kann. Darin findet Wonderboy Geschäfte, Kneipen, Wahrsager oder einfach Ungeheuer, die ihm ans Leder wollen.

Der Spieler muß Wonderboy durch das horizontal scrollende Szenario steuern und springend den umherlaufenden Gegnern ausweichen. Die Widersacher werden natürlich nach und nach immer stärker, so daß Wonderboy dringend Geld sammeln muß, um in den Geschäften seine Ausrüstung zu ergänzen. Wonderboy in Monsterland ist eines der bezauberndsten Spiele der letzten Monate. Obwohl es im Grunde nichts anderes ist als ein Jump-and-Run-Spiel, stecken so viele gute Ideen und nette Gags in diesem Programm, daß es immer wieder eine Freude ist, für eine Partie den Computer einzuschalten.

(hs)

AMIGA DOS Blitzlicht

Name: Wonderboy in Monsterland
Hersteller: Activision
Quelle: Fachhandel
Preis: 84,95 DM

Positiv:
- nette Spielfiguren
- Spielverlauf lustig und spannend
- gute Musik

Negativ:
- Grafik könnte besser sein





Flußfahrt mit Autoreifen. Toobin macht mindestens so viel Spaß wie ein Kurzurlaub

Toobin

Bif und Jet lieben es, reißende Flüsse mit aufgeblasenen Autoreifen zu befahren. Ein Spaß, der, solange man ihn in harmlosen Gewässern betreibt, recht ungefährlich ist.

Neben der tollen Landschaft, die an den Ufern der Flüsse bewundert werden kann, schätzen die beiden auch all die kleinen Abenteuer, die auf einer Flußfahrt passieren können. Da gibt es Angler, die ihre Ruten über den Fluß halten, abgebrochene Baumstämme treiben umher, Flaschen werden ins Wasser geworfen und einiges mehr, was mitunter noch schlimmer ist. Im Vergleich dazu nehmen sich Wasserfälle und Stromschnellen eher harmlos aus. Auf jedem Fluß haben sich die beiden eine bestimmte Strecke abgesteckt, ist diese durchfahren, geht es gleich auf dem nächsten Fluß mit anderen Hindernissen weiter.

Lust auf eine Flußfahrt? Nein? Das ändert sich spätestens nach einer Runde Toobin. Was hier begeistert und Spaß macht, ist zunächst einmal das originelle Spielgeschehen, das in dieser Form noch nicht dagewesen ist. Der Spieler steuert die Paddelbewegungen von Bif und Jet. Die Flüsse scrollen vertikal, so daß immer nur ein Teil zu sehen ist. Um nicht ganz wehrlos zu sein, können leere Dosen eingesammelt und Krokodilen und anderen Gefahren entgegengeschleudert werden. Bei Toobin kommt richtige Urlaubsatmosphäre auf, die durch die ausgesprochen originelle Musikuntermalung des Ganzen nur noch unterstrichen wird (der Calypso aus dem Titel ist absolut hörenswert). Erst nach Toobin weiß ich, wie viel Spaß eine Flußfahrt machen kann.

(hs)

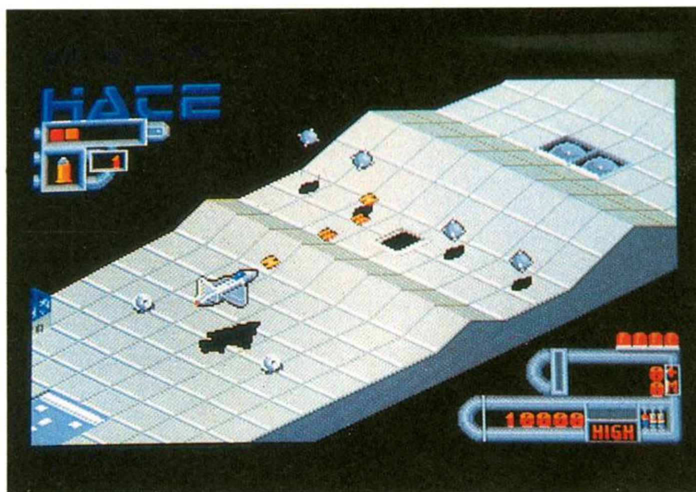
AMIGA DOS Blitzlicht

Name: Toobin
Hersteller: Domark
Quelle: Fachhandel
Preis: 64,95 DM

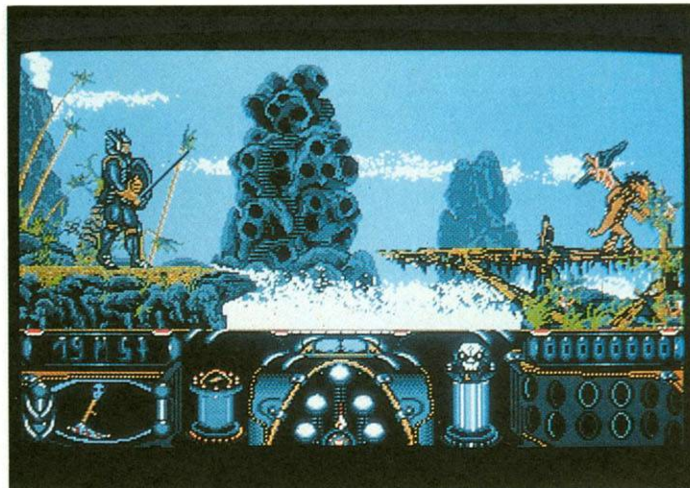
Positiv:
- originelles Spielgeschehen
- tolle Musik
- Continue-Funktion enthalten

Negativ:
- Grafik könnte besser sein





Über Stock und Stein geht es mit H.A.T.E.; hier darf auf jedem Terrain geballert werden



Zähes Schwertgefecht in fünf verschiedenen Zeiten: Knight Force

H.A.T.E.

Wer am Steuer eines Panzers, eines Kampfgleiters oder eines bewaffneten Wasserfahrzeugs eine gute Figur machen will, der muß üben, üben, üben. Und dazu gibt es H.A.T.E.

Diese Abkürzung steht für Hostile all Terrain Encounter, einem Simulator, in dem der Spieler mit allen möglichen Bodenfahrzeugen fahren und umgehen muß, während er dauernd simulierten Angriffen begegnet. Zu Beginn der Simulation muß der Spieler mit einem Kampfgleiter zum Einsatzort fliegen. Bereits auf dem Weg dorthin ist der Gleiter permanent gegnerischen Angriffen ausgesetzt. Während des Fluges kann die Panzerung des Gleiters ergänzt werden, so daß mehr als ein Treffer notwendig ist, um den Gleiter zu vernichten. Ist der Einsatzort erreicht, geht es in einem Panzer weiter. Neben der einfachen Standardbewaffnung, einer Doppelkanone, ist der Panzer ferner mit einer Art Granatwerfer ausgestattet, der es ihm erlaubt, über Bodenunebenheiten hinwegzufeuern.

Trotz vieler verschiedener Fahrzeuge, die gesteuert werden müssen, mag bei H.A.T.E. keine rechte Freude aufkommen. Die einzelnen Landschaften werden in bester 'Zaxxon'-Manier diagonal gescrollt und sind auch leidlich abwechslungsreich geraten. Allein das Spielgeschehen wird bald fade. Es gibt nur wenige unterschiedliche Gegner. Die einzelnen Levels unterscheiden sich kaum voneinander, und der Sound befleißigt sich des dumpfen Gehämmeres einer Rhythmusmaschine. Alles in allem ein Spiel, auf das man lieber verzichten sollte.

(hs)

AMIGA DOS Blitzlicht

Name: H.A.T.E.
Hersteller: Gremlin Graphics
Quelle: Fachhandel
Preis: 64,95 DM

Positiv:
- nettes Scrolling

Negativ:
- langweiliger Spielverlauf
- wenig abwechslungsreich
- nervige Musik
- magere Grafik



Knight Force

Tanya, Prinzessin von Belloth, wurde von dem bösen Magier Red Sabbath entführt, denn sie ist die einzige, die das Land Belloth noch vor dem Bösen retten kann.

Sie tragen den ehrenvollen Titel "Der Ritter des Donners"; nach Meinung der Bewohner Belloths der einzige Held, der die Prinzessin zurückbringen kann. Also legen Sie sich Ihre Rüstung an, prüfen das Ihnen vererbte Schwert, Steel Killer, noch einmal und machen sich auf den Weg.

Die Leute von Belloth kennen das Geheimnis der Zeitreisen, und Tanya befindet sich wahrscheinlich irgendwo in einer von fünf Zeitepochen. Ausgehend von einem Kreis aus fünf mystischen Hinkelsteinen kann der Spieler eine Zeitzone wählen, die erkundet werden soll. Der Weg in die Zukunft führt in eine Welt voller Satelliten und Roboter, die Ihnen nicht freundlich gesonnen sind. Schlagen Sie sich wacker, gelingt es Ihnen vielleicht, bis in die Höhle vorzudringen, in der die Prinzessin angekettet ist.

Wenn Sie sich ihr nähern, verschwindet sie und ein Magier taucht vor Ihnen auf. Einer von Red Sabbaths Schergen, der bekämpft werden muß. Gelingt es, ihn zu besiegen, können Sie zu dem Steinkreis zurückkehren und eine andere Zeitzone untersuchen.

Knight Force präsentiert sich als mageres Actionspiel, das kaum jemanden vom Hocker reißen kann, zumal hier nicht einmal zwischen den einzelnen Spielszenen gescrollt wird, sondern nur einfach umgeblättert. Ansonsten fällt die ungenaue Steuerung auf, die echten Spielspaß im Keim erstickt. Knight Force ist ein Programm, das man sich getrost sparen kann – bei dem gepfefferten Preis allemal.

(Robert Marz/hs)

AMIGA DOS Blitzlicht

Name: Knight Force
Hersteller: Titus
Quelle: Fachhandel
Preis: 84,95 DM

Positiv:
- Schwertkampfspiel mit fünf verschiedenen Szenarien

Negativ:
- Spielverlauf eintönig
- schwergängige Steuerung
- kein Scrolling




Eine ferne Wasserwelt ist der Hintergrund dieses sympathischen Spieles

TYPHOON THOMPSON

An einem Sommertag im Jahre 2124 verschwand ein intergalaktischer Passagierraumkreuzer. Die letzte Position, an der Flug 396 geortet wurde, war ein Meeresplanet im Omega-Sektor namens Aguar. Die Wahrscheinlichkeit, daß er den Fluten zum Opfer gefallen war, schien groß zu sein. Hoffnung auf Überlebende gab es nur noch vereinzelt, bis vor etwa zwei Monaten ein Signal empfangen wurde.

Drei Rettungsmannschaften machten sich seitdem auf, die Überlebenden zu bergen, aber das Schicksal ließ alle in die Hände der Sprites fallen, einer Gruppe von Lebewesen, die auf den Inselgruppen von Aguar hausen.

In dieser Situation werden Sie quasi als viertes Rettungsteam auf den Planeten gebracht. Das dortige Begrüßungsteam besteht aus zwei Wächtern, von denen Sie Instruktionen für den weiteren Ablauf der Aktion erhalten. So besteht die erste Aufgabe darin, vier magische Artefakte wiederzubeschaffen. Diese sind in weißen Gewölben versteckt, die ihrerseits im Zentrum von kleineren Inselgruppen liegen.

Die Inseln sind von den ungeselligen Sprites bewohnt, die Ihre Mission mit Hilfe von bojenartigen Fluggeräten zu stören versuchen. Wenn diese Geräte abgeschossen werden, bemühen sich die flinken Sprites, sich aus dem Staube zu machen. Bevor sie jedoch entkommen, sollten sie eingefangen und in einem Sack gesammelt werden. Sind alle Inseln von den Sprites befreit, können diese gegen die gesuchten Artefakte eingetauscht werden.

Für jede erledigte Mission erhält man von den Wachen Extrawaffen, die dringend benötigt werden, um weitere Aufträge zu bestehen. Am Schluß geht es darum, den letzten Überlebenden zu finden und sicher heimzubringen.

Das Fluggerät, in dem unser Held auf Rettungsaktion geschickt wird, sieht wie ein mit Raketenantrieb versehener Schwimmreifen aus. Er schwebt über die Meere und scheint nicht mit den besten Steuerungselementen bestückt zu sein, denn die Lenkmanöver sind sehr gewöhnungsbedürftig. Kriegt man einmal die Kurve, so bedarf es einiger Mühe, in die andere Richtung auszuweichen. Die Trägheit des Fluggeräts ist in manchen Situationen nicht ungefährlich. Übung ist hier angesagt, denn auch die flinken Sprites, die es einzusammeln gilt, sind wahre Kunstschwimmer.

Typhoon Thompson benutzt mehr als die Hälfte des Screens für die Sicht auf das Szenario. Außerdem sind einige Anzeigen bei der Lösung der Probleme hilfreich. Am sinnvollsten erscheint eine Art Kompaß, mit dem man die gesuchten Inselgruppen schnell ausfindig machen kann.

Was an der grafischen Darstellung des Spiels besonders auffällt, ist die Nähe zu comicartiger Zeichnung. Das Programm scheint bewußt von der pseudorealen Darstellungsweise wegzugehen. Die Sprites sind ziemlich klein gehalten. Eine äußerst detaillierte Animation war so möglich. Das Ergebnis sind Bewegungsabläufe, die einfach sehenswert sind. Sie stecken voller Tempo und Humor.


Typhoon Thompson: kleine Sprites, aber blendende Animationen

Allein die Szene in der Thompson von seinem Vorgesetzten rüde davon überzeugt werden muß, daß er der richtige Mann für diesen Auftrag ist, ist ein echter Augenschmaus. Nach diesem kleinen Zeichentrick-Intro folgt eine Sequenz, in der der Spieler mit den einzelnen Komponenten des Spiels bekannt gemacht wird. Auch was hier geboten wird, geht über das Übliche hinaus.

Was die Bedienung des Spiels anbetrifft, brauchen keine langen Lernprozesse gefürchtet werden. Dank der hohen Motivation wagt man immer wieder 'nur noch ein Spielchen'. Wenig später ist die Steuerung Routine, und man kann sich auf die wesentlichen Spielkomponenten konzentrieren. Für das Erledigen der gestellten Aufgaben gilt das nicht, denn hier gehört neben dem geschickten Umgang mit dem Joystick auch ein wenig Köpfchen dazu.

Beeindruckend an Typhoon Thompson ist meiner Meinung nach die Art der grafischen Präsentation. Man hat nie das Gefühl jener Bierernsthaftigkeit, die manche Produkte schwer im Magen liegen lassen. Typhoon wirkt durch seine Lockerheit ungemein sympathisch. Wenn hier auch mit Laserkanone oder anderen Waffen geschossen wird, Verletzungsgefahr besteht für keinen.

(jf)

AMIGA DOS Blitzlicht

Name: Typhoon Thompson
Hersteller: Broderbund
Quelle: Fachhandel
Preis: 74,95 DM

Positiv:
- tolle Animationen
- sympathisches Spielgeschehen
- gute Grafik

Negativ:
- knödelige Soundeffekte

AMIGA DOS URTEIL GUT

GRAFIK	SOUND	MOTIVATION
7	4	8

AMIGA DOS – Spieletips

Es ist wieder Playtime rund um den Amiga. Bunt gemischt präsentieren sich unsere Spieletips heute. Von Ports of Call bis King's Quest lassen wir kein Genre aus. Und als besonderen Leckerbissen haben wir eine Karte der Wüstenstadt, die in 'It came from the Desert' der Schauplatz einiger unheimlicher Ereignisse wird.

Die Handelssimulation Ports of Call ist auf dem Amiga weit verbreitet und hat viele Freunde. Wir wollen die grundlegenden Verhaltensweisen, die zum Erfolg führen, noch einmal auflisten.

Ports of Call

Die besten Heimathäfen sind: San Francisco, New York, Housten und Vancouver. Am Anfang ruhig Schiffe der mittleren Preisklasse kaufen, sie bekommen bessere Frachten. Ständig auf den Tank achten und wenn das Öl billig ist (unter einhundert Dollar) auftanken. Schiffe in Ländern der dritten Welt reparieren lassen. Der Zustand eines Schiffes sollte 75% nicht unterschreiten. Lange Strecken fahren, kurze rentieren sich erst, wenn man mindestens drei Schiffe besitzt. Waffen, Chemikalien und Elektronik bringen am meisten ein. Nicht immer die gleiche Strecke fahren, auch wenn andere zunächst weniger einbringen. Wenn man mehr als fünf Schiffe besitzt, sollte man feste Routen einrichten. Besonders günstig sind dabei: USA-Europa und Afrika, Europa-Asien, Asien-Australien. Schwimmende Schrotthaufen sollten Stürme meiden. Bei zwielichtigen Geschäften immer erst ablehnen, dann überlegen, das Risiko ist groß. Mindestens alle zwei Monate ins Büro schauen, Gelder könnten sonst veruntreut werden. Vorsicht vor Angriffen im Persischen Golf!

Von den Weltmeeren und seinen Supertankern begeben wir uns nun ins Weltall mit den nicht minder großen Raumschiffen. Genauer gesagt verfolgen wir unseren Freund Roger Wilco bei seinen Versuchen, die Galaxis zu retten.

Space Quest I

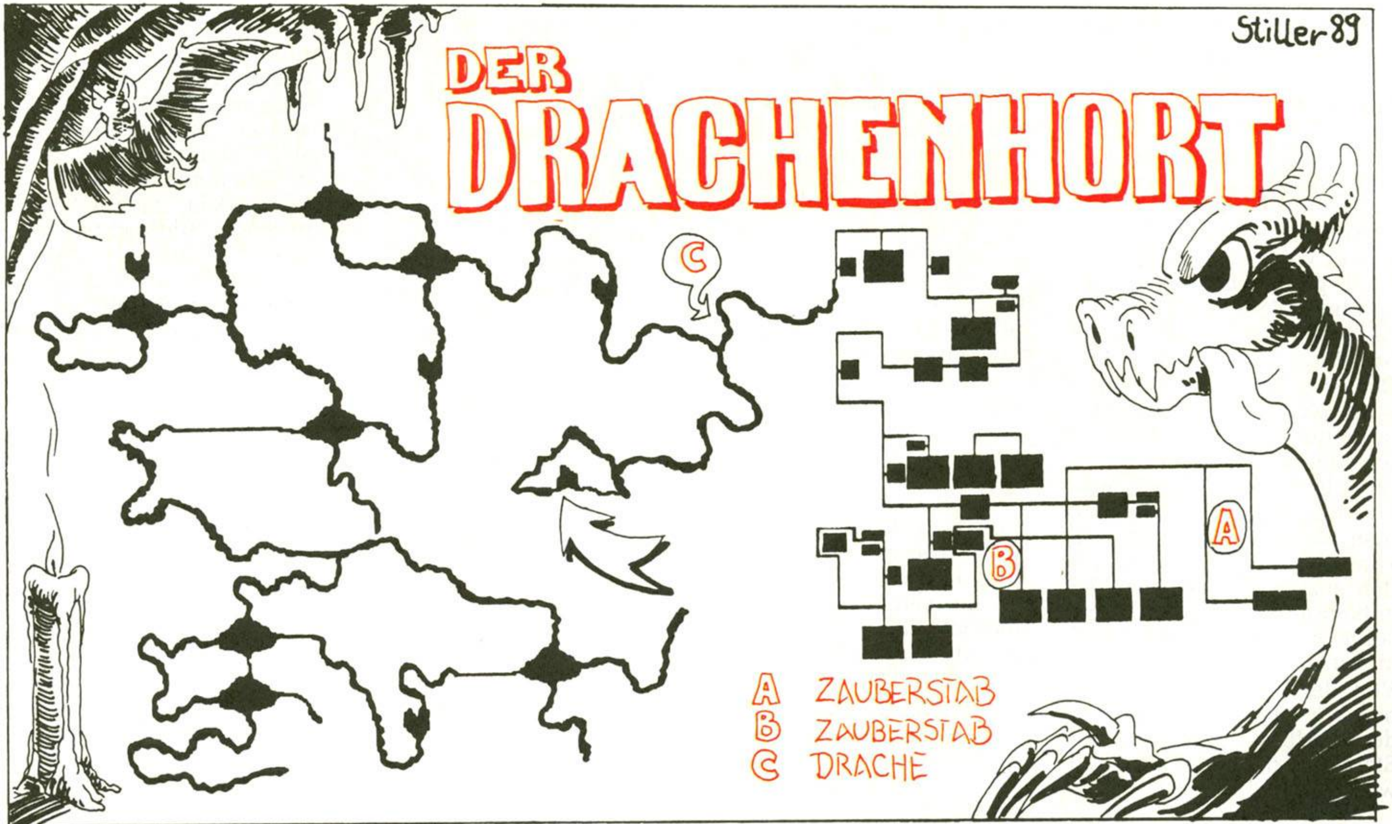
Zunächst muß Roger einmal von dem von den Sariens gekaperten Raumschiff flüchten. Es patroullieren ständig Wachen im Schiff. Sollte Roger also Schritte hören, lassen wir ihn sofort einen Raum oder Fahrstuhl aufsuchen. Von seinem Startpunkt aus bewegen wir Roger zunächst nach links ins Archiv, in das auch schon bald ein verwundeter Wissenschaftler kommt und uns den Namen des Datapacks verrät, das wir für unsere Mission brauchen. Der Name ist 'astral body'. Die Daten allerdings nicht vom Computer anzeigen lassen, sondern nur den Datenträger nehmen und den Raum verlassen. Für seine Flucht braucht Roger eine Keycard höchster Priorität, die er findet, wenn er seine toten Kameraden mit 'search body' untersucht. Dann benutzen wir den Fahrstuhl in den unteren Sektor des Schiffes. Wir halten uns rechts, gehen zum Steuerpult und drücken 'Bay open'. Wieder zurück nach rechts zum Fahrstuhl, den wir mit der Keycard öffnen und benutzen. Nun muß man die Airlock (Schleuse) öffnen und aus den hinteren beiden Türen im oberen Bildrand den Raumanzug und den Translator nehmen. Jetzt begeben wir uns durch die Schleuse in den Hangar, holen mittels Knopf den Shuttle hoch und steigen ein. Nachdem wir die Türen geschlossen und den Gurt angelegt haben, drückt Roger auf Power, und ab geht's. Nach dem Verlassen des Schiffes braucht nur noch der Autopilot eingeschaltet zu werden, und einer Bruchlandung auf dem Planeten steht nichts mehr im Wege. Wir nehmen eine Glasscherbe der Cockpit-Scheibe, das Messer und die Flasche mit Wasser, die aus dem Survival Kit herausgefallen sind.

Auf geht's nach rechts in die Wüste. In dem Durchgang nach rechts führt ein verdeckter Weg in eine Höhle, in der sich Orat befindet. Wir schleudern ihm das Wasser an den Kopf und nehmen ein Stück von ihm mit (das, welches rechts außen liegt). Weiter führt uns der Weg nach rechts, den Aufgang zum Plateau hinauf und über die Brücke, wo wir hinter dem Felsen Aufstellung nehmen. Wir warten auf den Sarien Droiden, und wenn er unter uns ist, lassen wir den Stein auf ihn fallen. Jetzt ersteigt Roger die Plattform und stellt sich genau zwischen die beiden Steinsäulen. Ein Fahrstuhl befördert ihn dann nach unten. Roger nimmt den Stein, der sich direkt neben dem Fahrstuhl befindet, an sich und macht sich auf den Weg. An dem Gitter, unter dem sich ein Alien befindet, drückt er sich ganz am oberen Rand vorbei. Jetzt lassen wir Roger den Stein auf den Geysir werfen, worauf dieser außer Gefecht gesetzt ist. Die Laserbarriere überwindet er, indem er den Glassplitter in die Strahlen hält. In der Tropfsteinhöhle sollte er sich vor herabfallenden Tropfen in acht nehmen. Jetzt schalten wir den Translator ein und begeben uns in den nächsten Raum, in dem wir den Orat-Rest tauschen. Die Treppe hinauf, das Angebot mit dem Gleiter annehmen und ihn mit dem Schlüssel starten. Jetzt untersuchen wir den Computer im hinteren Teil des Gleiters, schieben die Cartridge ein und merken uns vor allem den Selbstzerstörungs-Code. (6858). Jetzt sollte auf jeden Fall zwischengespeichert werden, da die nachfolgende Actionsequenz ziemlich schwer ist. In der Stadt schlagen wir das erste Angebot für den Gleiter aus, das zweite nehmen wir an. Wieder speichern und ab in die Bar, wo so lange gespielt wird, bis Roger die Bank gesprengt hat.

Danach begeben wir uns zum Schiffshändler und kaufen das Raumschiff in der linken hinteren Ecke des Platzes. Nachdem wir uns auch noch einen Droiden zugelegt haben, der den Gleiter steuern kann, laden wir diesen ins Schiff ein, und los geht's. Unser Zielsektor ist, wie wir in der Bar erfahren haben, der Sektor HH. In der Nähe des Sarienschiffs trennen wir uns von unserem Droiden und fliegen mit dem Jetpack zur Schleuse, die wir mit dem kleinen Hebel öffnen. Die innere Schleuse öffnet ein kleiner Roboter. Im nächsten Raum geben Sie folgende Befehlssequenz ein: Open Trunc, get off Jetpack, put jetpack into Trunc, close Trunc, move Trunc und schieben die Kiste unter den Ventilator. Auf die Kiste mit climb trunc, und den Ventilator mit dem Messer öffnen. Jetzt durch die



The Fairy-Tale-Adventure – Lageplan der Stadt Marnheim



The Fairy-Tale-Adventure – Lageplan der Drachenhöhle

Schächte und den Ausgang mit kick vent öffnen. Schnell hinein in die Waschmaschine. Jetzt lenken wir Roger, der nun in Uniform stecken sollte, durch die Tür, benutzen den linken Fahrstuhl, halten uns links und gehen wieder in einen anderen Fahrstuhl. Nach oben und rechts. Die große Halle durchqueren wir auf der Balustrade und begeben uns in die Waffenkammer. Dem Droiden geben wir die ID-Karte aus der Uniform. Wenn er weg ist, um unsere Waffe zu holen, müssen wir schnell die Granate nehmen, da der Roboter bald zurückkommt, und es nicht bemerken darf. Roger geht jetzt zurück in die Halle und wirft die Granate auf die Wache. Jetzt zum Fahrstuhl, nach unten und schneller als die Sariens schießen. Unten durchsuchen wir die tote Wache und öffnen den Schutzschirm des Stargenerators mit der Remote Controll. In die Maschine jetzt den Selbstzerstörungs-Code eingeben und nichts wie weg. Wir rennen in den Raum vor der Wäscherei, benutzten den rechten Fahrstuhl, steigen in den kleinen Gleiter ein und drücken auf launch. Wenn Sie jetzt das Raumschiff verlassen, sehen Sie, wie es sich in seine Bestandteile auflöst. Roger kann aufatmen, er hat es geschafft, die Galaxis zu retten.

Wir führen einen Riesensprung vom Weltraum in eine fantastische Märchenwelt durch, genauer gesagt, führen uns unsere nächsten Tips in das Land Holm. Feary Tales sind angesagt. Zunächst einmal eine Komplettlösung, anschließend folgen noch ein paar allgemeine Hilfen:

Fairy-Tale-Adventure

Auf der Suche nach den fünf goldenen Statuen begeben wir uns zunächst einmal in das Schloß der guten Fee. Diese überreicht uns dann auch schon die erste der fünf Statuen und einiges dazu. Von der Fee geht es auf dem schnellsten Wege zu der Höhle mit dem Drachen, den wir bekämpfen und hoffentlich auch töten. In seinem Hort hat er einen Zauberstab, den wir an uns nehmen. Unser nächstes Ziel ist Marnheim, wo wir eine Audienz beim König erhalten. Dieser gibt uns dann auch gleich den Auftrag, seine entführte Tochter wiederzubeschaffen. Wir schlagen also eine östliche Richtung ein und begeben uns zum Wachturm. Am Strand liegt eine Muschel, die wir natürlich an uns nehmen. Im Grimwood liegt an einer Feuerstelle eine weitere der goldenen Figürchen, wir wagen uns tiefer in den Wald und stoßen zu einer alten Hexe vor, die uns ebenfalls alles andere als freundlich gesonnen ist. Nachdem auch sie keine Gefahr mehr darstellt, finden wir in ihrer Höhle ihr goldenes Lasso. Der Steinkreis südlich der Wüste ist unser nächstes Ziel. Dort gehen wir ans Meeresufer und rufen mit der Muschel eine Schildkröte herbei. Auf dem Rücken der Schildkröte läßt es sich bequem reisen. Sie trägt uns zur Schwaneninsel, auf der wir mit Hilfe des goldenen Lassos den Schwan einfangen und bändigen können. Der Schwan wird uns für den Rest des Adventures seine Schwingen zur Verfügung stellen und uns überall hintragen, wo immer wir wollen. Das erste Ziel, das wir fliegend ansteuern, ist Helmsaths Gruft. In den Tiefen des Labyrinths liegen die dritte Statue und ein alter Knochen. Den Knochen tragen wir auf den Friedhof an der Straße nach Marnheim. Hier taucht dann auch bald ein Skelett auf, das uns im Austausch mit dem Knochen einen Kristallsplitter überreicht. Ist auch dieses erledigt, fliegen wir in die Berge zum Sonnenstein. In dessen Nähe wird des Königs Tochter in einem Verlies gefangengehalten. Haben wir sie gerettet, finden wir uns auch schon in Marnheim wieder und erhalten als Lohn die vierte goldene Statuette. Die fünfte und letzte liegt im alten Schloß versteckt. Jetzt steht einem Abstecher nach Azal nichts mehr im Wege. Das Erste, was wir hier finden, ist eine Rose mit besonderen Eigenschaften, die noch sehr nützlich sein werden. In der Zitadelle der Verdammnis kommt es dann zum großen Schlußkampf.

Einige Kleinigkeiten, wie man in Holm besser über die Runden kommt, bleiben noch nachzutragen:

Der Friedhof bei Marnheim eignet sich dazu, Punkte und magische Utensilien zu sammeln, wenn man nicht mit einem Dirk kämpft. Dazu stellt man sich hinter den Zaun und kämpft von dort aus. Die Steinkreise sind Transporter,

kosten aber pro Transport einen blauen Stein. Verletzungen heilt man mit Vials.

Eingänge zu Höhlen und ähnlichem findet man mit Totems.

Mit Schlüsseln sollte man sparsam umgehen, da man sie nur einmal benutzen kann.

Wenn man Bettlern und Wanderern von Zeit zu Zeit etwas Geld gibt, erhöht dies den Kindnes-Wert, und man hat es später beim Verhandeln leichter.

Der Ring läßt die Feinde für kurze Zeit erstarren, dies ist besonders in der Drachenhöhle von großem Nutzen.

Im Grimwood sollte man sich vor dem Sumpf in acht nehmen, sonst versinkt man und kommt in ein Labyrinth, aus dem man nur schwer wieder entkommt. Gegen die Hexe sollte man den Magic-Wand einsetzen.

Von der Märchenwelt begeben wir uns jetzt wieder in den Weltraum. Hier wartet Starglider II darauf, gelöst zu werden.

Starglider II

Das erste, was Sie tun sollten ist, in der Zentrale den Bau einer Neutronenbombe in Auftrag zu geben. Die dazu benötigten Teile müssen Sie selbst heranschaffen. Man erhält sie entweder durch Handel mit anderen Depots oder kann sie auf einigen Planeten und Monden direkt einsammeln. Spätestens wenn alle sieben Projector Bases fertiggestellt sind, sollte man diese mit Hilfe der Bouncing Bombs von Castron zerstören. Sonst wird die Egron Space Station zu früh fertiggestellt, und für den Heimatplaneten gibt es keine Rettung mehr. Sind alle Gegenstände und auch der Professor in der Zentrale, kann mit dem Bau der Bombe begonnen werden. In der Zwischenzeit sollten Sie die Sentinels, die die Egron Space Station bewachen, mit Hilfe von mindestens acht Homing Missiles vernichten. Sind die Sentinels alle fort, kann man sich die Bombe von Apogee abholen und die Station vernichten.

Auch zu Starglider II haben wir noch einige allgemeine Tips:

Auftanken kann man am besten im Asteroiden-Feld zwischen Apogee und Millway.

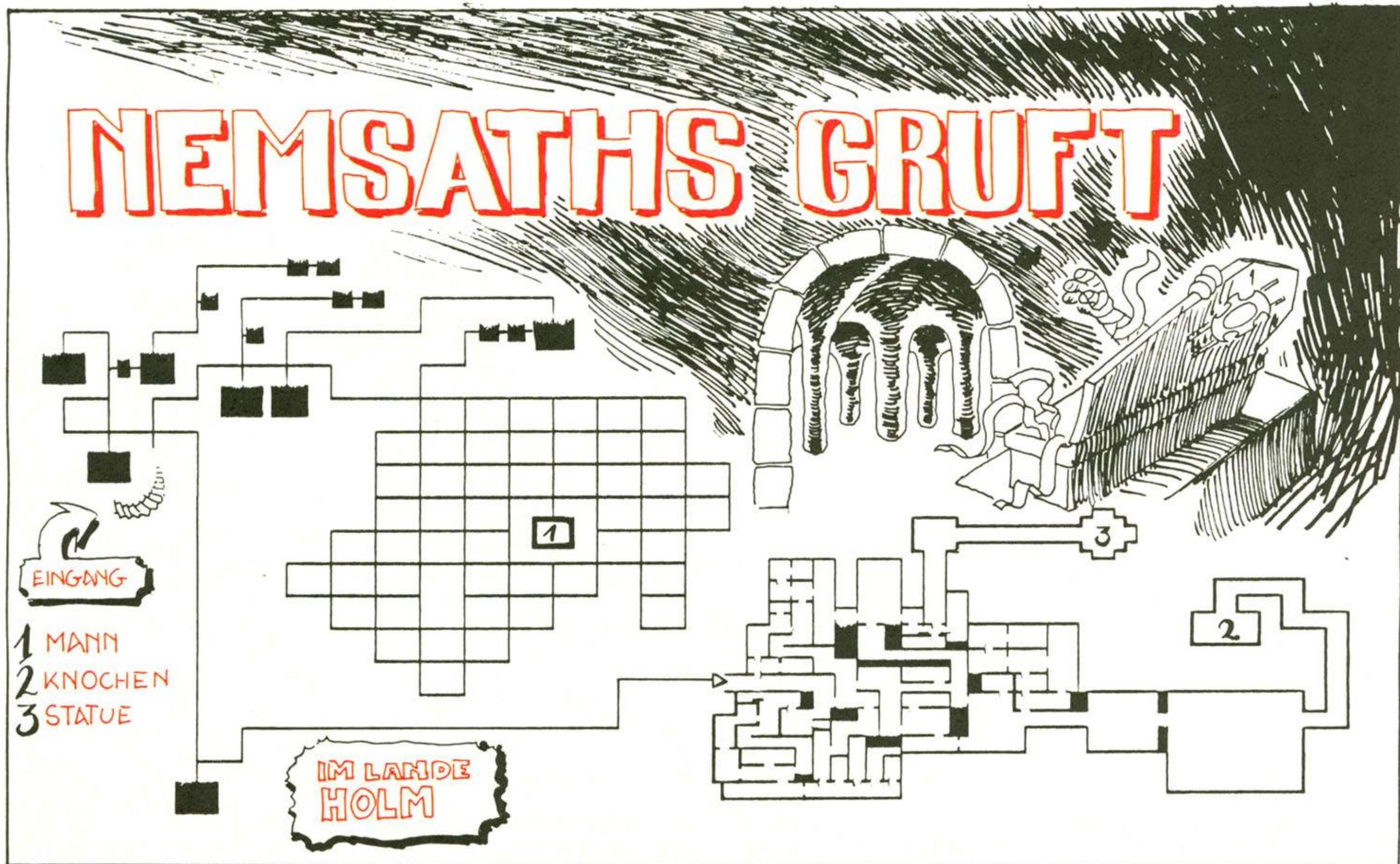
Will man sich im All orientieren, so empfiehlt sich Computer Visual Analysis, womit man den Namen des eingepeilten Planeten erfährt.

King's Quest I

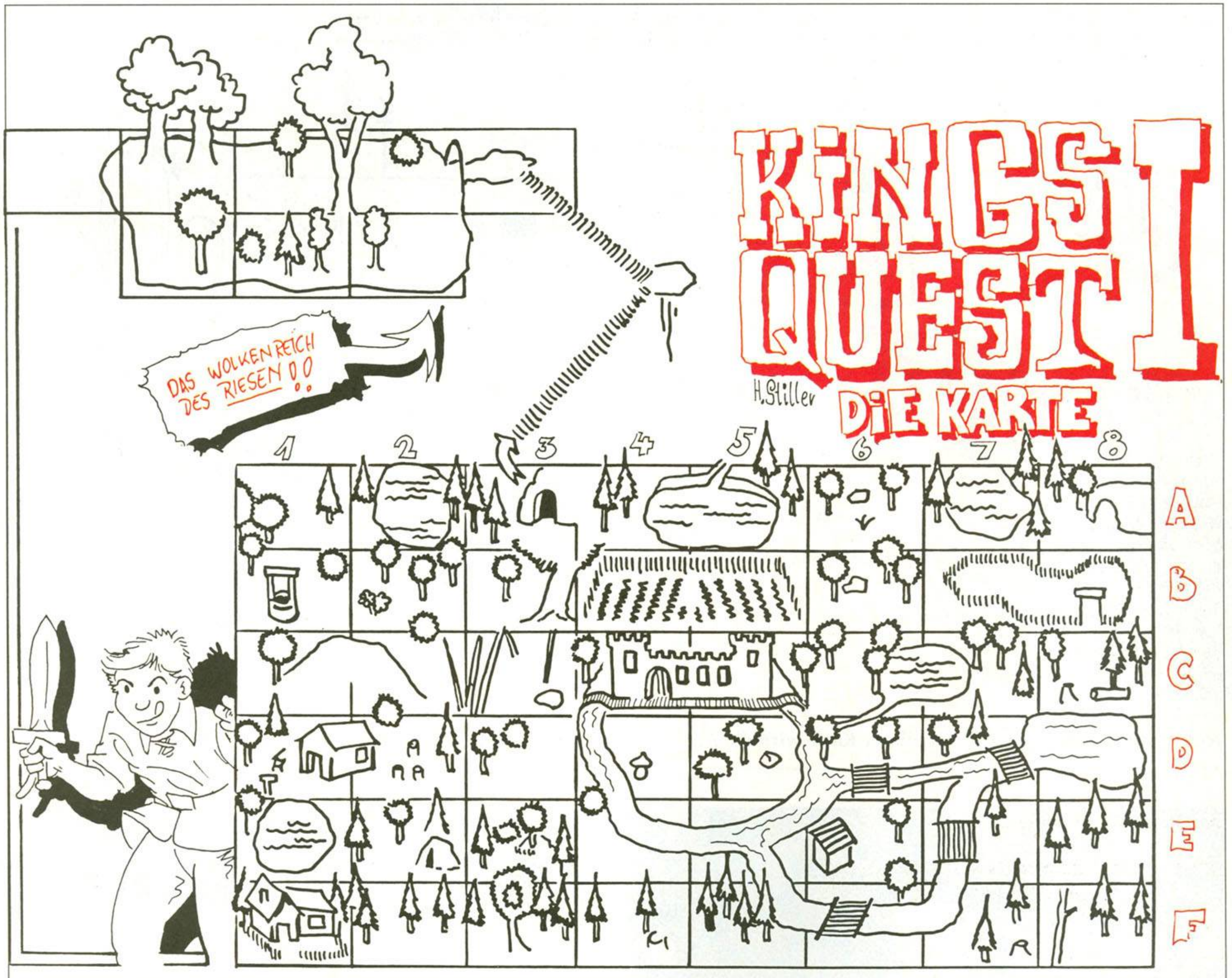
Als allererstes suchen wir das Schloß auf und begeben uns zum König, von dem wir unseren Auftrag erhalten, nachdem wir mit ihm gesprochen haben (Speak King). Danach gehen wir zu dem Stein in Sektor G4, bewegen ihn und schauen in das sich auftuende Loch. Mit dem Befehl “Take Dagger“ erhalten wir ein Schwert. Die nächste Etappe ist der Gold-Ei-Baum (G3), den wir ersteigen. Wir lassen uns, nachdem ein Ei in unseren Besitz übergegangen ist, wieder herunterfallen. Dann gehen wir zum Troll an der Brücke, geben ihm das Ei und passieren. Beim alten Gnom (B6) müssen wir dreimal den Namen raten, nach dem wir zu ihm gesprochen haben. Den Schlüssel nehmen wir an uns. Der Schlüssel paßt zu der Tür in dem Berg (G2), die sich nun aufschließen und öffnen läßt. Die Treppe auf der rechten Seite vorsichtig ersteigen, aber aufgepaßt, es besteht Absturzgefahr! Auf dem Absatz erscheint ein Gnom, der aber harmlos ist, wenn man nicht mehr mitführt, als wir bis jetzt gefunden haben. Ansonsten nimmt er es uns weg. Oben angelangt, befindet man sich über den Wolken, wieder aufgepaßt: an den Rändern besteht auch Absturzgefahr. Wir begeben uns von Feld GX9 über GX11 nach GX10. Im nächsten Bild rechts befindet sich ein Riese, den man auf zwei Arten überlisten kann: Entweder man geht vom oberen Drittel des Bildes GX10 nach rechts zu GX8, wo wir uns schnell hinter einem Baum verstecken und dort bleiben. Nach einiger Zeit schläft der Riese dann ein. Eine andere Möglichkeit besteht darin, ihm etwas mit der Fiedel aus der Hütte des Holzfällers (E5, F5) vorzuspielen, worauf dieser dann auch einnickt. Wenn der Riese schläft, nehmen wir ihm seine Kiste ab.

Jetzt verlassen wir den Riesen wieder und kehren über die Treppe zurück. Draußen schließt sich dann die Tür wieder, und die erste Aufgabe ist gelöst.

Jetzt gehen wir zu dem Bild B2. Falls in H2 ein Elf erscheint, sprechen wir mit ihm und erhalten einen Ring, der unsichtbar macht. In B2 nehmen wir etwas Klee, mit dem



The Fairy-Tale-Adventure – Lageplan von Nemsaths Gruft



Die King's-Quest-Karte, für alle, die bei diesem Spiel keinen Plan haben

wir uns nach B3 begeben. Dieses Bild betreten wir solange von allen Seiten immer wieder, bis eine gute Fee erscheint, die uns durch einen Zauberspruch vor der Hexe beschützt. Mit diesem Schutz-Zauber begeben wir uns auf dem schnellsten Wege ins Hexenhaus im Bild E1. Mit 'open door' betreten wir dieses. Wir sollten uns auf jeden Fall beeilen, da der Zauber außerhalb des Hauses nur kurz anhält. Wir öffnen den Schrank (cabinet) und nehmen den Käse aus ihm heraus. Haben wir die Nachricht, die auf dem Tisch liegt, genommen und gelesen, verlassen wir das Haus wieder. Und betreten und verlassen Bild D2 solange, bis der Vogel erscheint. Ist er da, springen wir solange hoch, bis er uns mitnimmt. Er befördert uns auf eine Insel, auf der wir uns nach links bewegen. In diesem Bild finden wir einen Pilz, der in unseren Besitz übergehen sollte (take mushroom). Wir kehren nach rechts in das vorherige Bild zurück und springen mit 'jump hole' in das Loch. Richtung Süden, von dort weiter links. Die Ratte, die den Weg versperrt, füttern wir mit dem Käse, worauf sie verschwindet. Jetzt öffnen wir die Tür am linken, oberen Rand. Haben wir jetzt den Klee dabei, passiert nichts. Fehlt er uns aus irgendeinem Grunde, sterben wir leider. Haben wir also überlebt, gehen wir unermüdlich weiter zu dem Zwergenkönig. Wir nehmen seinen Schild und verlassen das Bild nach links. Links befindet sich jetzt eine kleine Öffnung. Wir gehen bis dicht davor und essen den Pilz. Kurzfristig werden wir jetzt so klein, daß wir durch die Öffnung passen. Die Wirkung hält aber nicht sehr lang an, so daß wir uns ein wenig beeilen müssen. Wir sind jetzt wieder auf dem Festland in Bild F6. Nachdem wir nun also auch den zweiten Teil der Aufgabe gelöst haben, gehen wir den dritten und letzten an:

Wir gehen zum Brunnen, der sich in Bild E3 befindet, und klettern in den Eimer, worauf dieser uns langsam bis zur Wasseroberfläche hinunterfährt. Jetzt gibt man folgende Befehlssequenz ein: jump water, swim, cut rope, dive. Gibt man etwas anderes ein, ertrinkt man. Im nächsten Bild tauchen wir nach links unten durch die Öffnung zum feuerspeienden Drachen. Wir löschen sein Feuer mit 'throw water', worauf dieser die Höhle nach links verläßt. Wir nehmen den magischen Spiegel und verlassen das Bild nach links. Jetzt nur noch zum Schloß und hinein zum König.
Bow King ist unser letzter Befehl, worauf dieser zufrieden entschlummert. Von jetzt an sind wir der "King of Daventry" und besteigen als solcher den Thron.

It came from the Desert

Mit unserer Wüstenkarte sollte die Jagd auf die Monsterameisen in Zukunft kein zu großes Problem mehr darstellen. Damit kommen wir auch schon zum Ende der Helpline dieser Ausgabe. Wir hoffen, daß auch für Sie etwas dabei war. Ansonsten wünschen wir noch viel Spielspaß mit Ihrem Amiga.

(Robert Marz/hs)

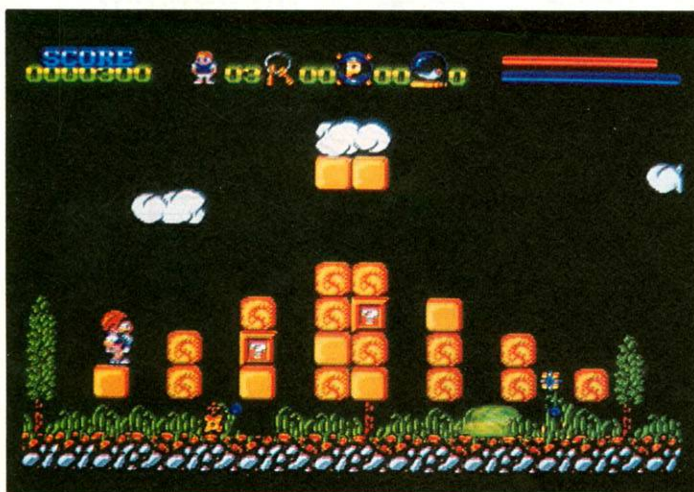
IT CAME FROM THE DESERT

IS WAS?

DIE STADT LIZARD BREATH

1 ORE PLANT
2, 3, 4 PUMPSTATIONEN
5 WILLIAMS FARM
6 RADIO STATION
7 LABOR
8 HILPERS FIELD
9 NEPTUNS HALL
10 COOKS FARM
11 DRIVE IN
12 J. D's FARM
13 QUARRY
14 ARMY
16 HOME
17 FUEL STATION

Wo ist was in Lizard Breath? Mit dieser Karte wird die Ameisenhatz ein wenig leichter



Und dann haut er mit dem Hämmerchen das Monster – Axel auf der Suche nach seiner Freundin

Axel's Magic Hammer

Nilmerg, seines Zeichens groß, grün, böse und Drache, hat Axels Freundin entführt. Doch damit, daß Axel, der Held, seinen magischen Hammer auspackt und bei der Rettung seiner Freundin alles kurz und klein haut, konnte er nicht rechnen.

Durch acht Länder muß sich Axel schlagen, ehe er Nilmergs Festung erreicht hat. Diese Regionen sind mit allerlei unfreundlichem Getier bevölkert, die von Nilmerg gedungen wurden, Axel aufzuhalten. Der ist jedoch nicht zu bremsen und schlägt, inklusive großer Felsbrocken, alles kurz und klein. Mit gutem Grund, denn in den Steinen verbergen sich oft wertvolle oder nützliche Gegenstände, die Axels Aufgabe ein wenig leichter und das Leben seiner Feinde ein wenig schwerer machen. Der Weg zu Nilmergs Festung ist aber leider nicht ausgeschildert, und mancher Irrweg ist noch mit ein paar tödlichen Fallen gewürzt. Mitunter muß Axel sich seinen Weg mitten durch eine geschlossene Felswand bahnen, um dem Hort des Bösen näher zu kommen.

Axel's Magic Hammer wirft viele Elemente aus bekannten und beliebten Hüpf- und Sammelspielen in einen Topf und macht daraus einen ausgesprochen gut spielbaren Eintopf. Der Spieler kann Axel durch alle Räume steuern und durchaus zu einem schon durchwanderten zurückkehren. Spätestens im zweiten Level ist das auch bitter nötig, da hier viele offensichtliche Irrwege existieren und einige, die erst gefunden werden müssen. Abgesehen davon gibt es auch versteckte Höhlen, die über und über mit Schätzen gefüllt sind. Axel's Magic Hammer darf zweifelsohne zu den besten Hüpf- und Sammelspielen für den Amiga gezählt werden.

(hs)

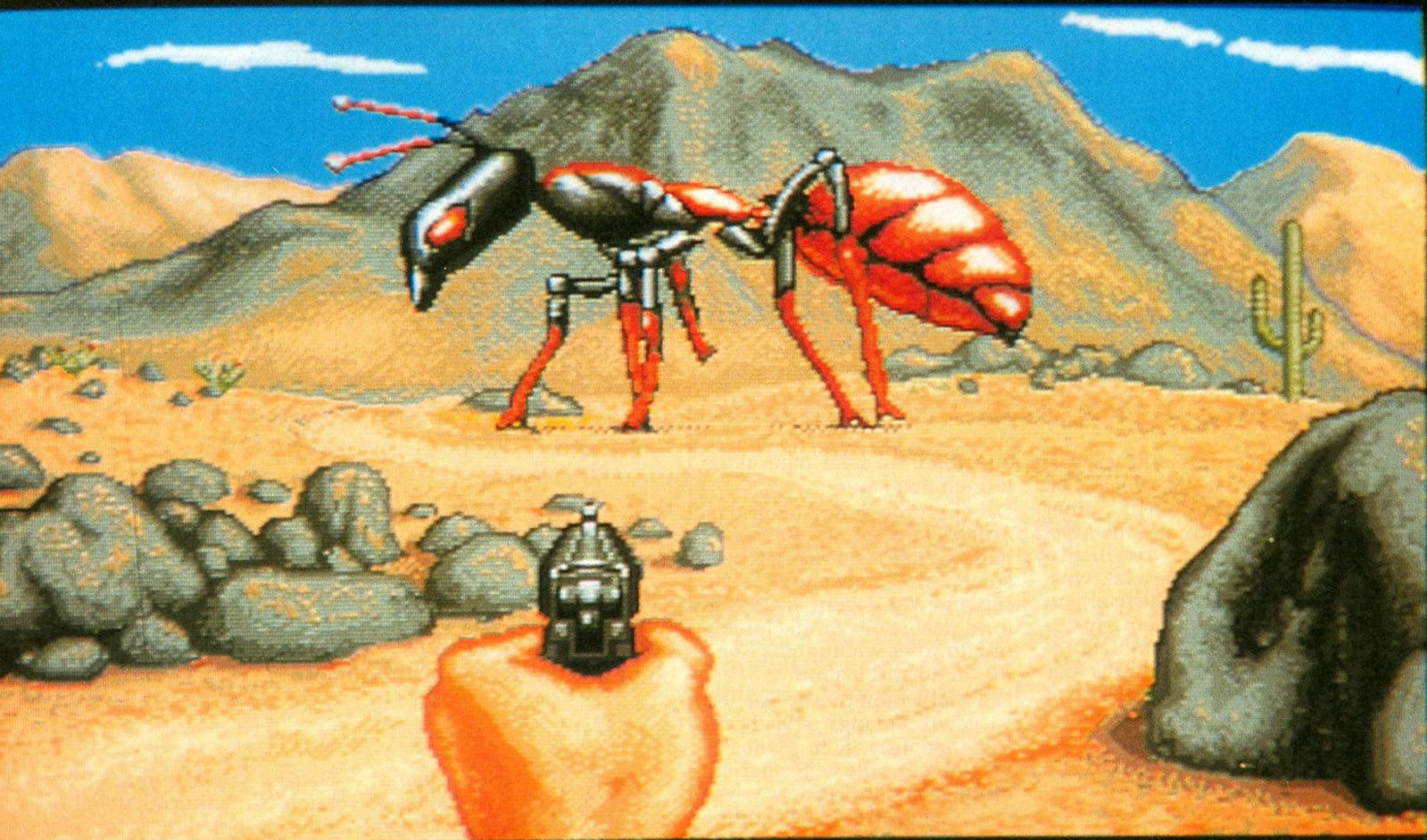
AMIGA DOS Blitzlicht

Name: Axel's Magic Hammer
Hersteller: Gremlin Graphics
Quelle: Fachhandel
Preis: 64,95 DM

Positiv:
- putzige Grafik
- Adventuretouch
- probate Steuerung

Negativ:
- mäßige Musik





IT CAME FROM THE DESERT

1951. Ein Meteorit geht nahe des amerikanischen Städtchens Lizard Breath nieder. Was hat es mit diesem Meteoriten auf sich, und welche Auswirkungen auf die Lebewesen in diesem Landstrich hat der Absturz? Diese Fragen gilt es aufzuklären. Dabei werden Sie mit haarsträubenden Geschehnissen konfrontiert, wie sie in Ihren schlimmsten Alpträumen nicht hätten vorkommen können.

Kennen Sie den Film Formicula von Jack Arnold, in dem Riesenameisen die Bevölkerung bedrohen? Das Adventure It came from the Desert basiert auf diesem Klassiker. Zur Geschichte:

Sie übernehmen die Rolle eines geologischen Wissenschaftlers, der mit der Untersuchung von Gesteinsproben beauftragt ist. Geez, der alte Mann, den Sie beauftragt haben, Ihnen einige Exemplare zu besorgen, klopft eines Tages an Ihre Tür und übergibt Ihnen einige der gewünschten Proben.

Formicula läßt grüßen

Nach kurzer Untersuchung fällt dabei ein seltsamer Stein auf, dessen Zusammensetzung unbekannt ist. Nachdem Sie Geez nach der Herkunft des seltsamen Steines gefragt haben, beginnt die Geschichte interessant zu werden. Ein Besuch bei der Fundstelle führt Sie zu einer nahegelegenen Farm, dessen Besitzer Sie über den Tod einer seiner Kühe aufklärt.

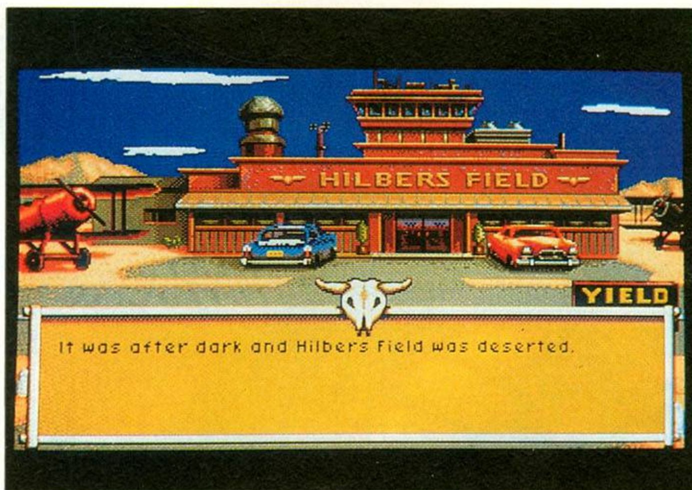
Schon der Besuch auf besagter Farm bringt Sie fast an die Grenze der Belastbarkeit Ihrer Nerven, denn plötzlich taucht eine mutierte Ameise auf, größer als ein Haus! Kurzerhand greifen Sie nach Ihrer Pistole, um dem Ungetüm den Garaus zu machen. Gibt es noch mehrere Riesenameisen? Was ist der Grund für diese Mutation?

Dies alles sind Fragen, auf die Sie in diesem Grafikadventure stoßen werden. Der Spieler kann seine Spielfigur in der Art beeinflussen, indem er sie an bestimmte Orte führt, um dort spezielle Informationen zu erlangen oder bestimmte Tätigkeiten auszuführen. Dies geschieht über einen Pfeil, der durch den Joystick bewegt wird. Über Dialogboxen können an den Orten dann Personenbefragungen durchgeführt oder Entscheidungen getroffen werden.

Aber auch Actioneinlagen sind bei It came from the desert nicht vergessen worden: Sollte es zu einer Konfrontation mit einer Riesenameise kommen, so erscheint eine Pistole auf dem Bildschirm, die durch Joystickbewegungen geführt wird.

Auch Action kommt nicht zu kurz

Dabei kommt es darauf an, den Ameisen die Fühler abzuschießen. Dies ist gar nicht einmal so einfach, denn diese Biester bewegen sich relativ schnell auf Sie zu und, da Ihr Munitionsvorrat nur begrenzt ist, muß so manches Mal das Magazin gewechselt werden. Außerdem ist die Umsetzung der Joystickbewegung zur Bewegung des Schußarmes doch



Nach einem Meteoritenabsturz gehen seltsame Veränderungen in dem kleinen Wüstenstädtchen Lizard Breath vor. Können Sie das Geheimnis lüften?

recht langsam. Sollte es vorkommen, daß Sie (egal, ob durch den Angriff einer Ameise oder durch eine Messerstecherei mit zwielichtigem Gesindel) verletzt wurden, so erwachen Sie in einem Hospital. Aus diesem Hospital ist nur ein Entkommen, indem Sie die Anordnung der Krankenschwester befolgen (Zeitverlust) oder aber einen Fluchtversuch unternehmen. Dieser Fluchtversuch ist jedoch auch nicht so einfach, da die Krankenschwestern und Ärzte Sie an dieser Aktion hindern wollen. Im Spiel erinnert diese Sequenz ein wenig an Gauntlet. Zur Zeiteinteilung noch ein kurzes Wort:

Jeder kann sich bestimmt vorstellen, daß Eile Not tut, denn je länger man braucht, die Ameisen zu vernichten, desto sehr vermehren sich diese Biester auch. Außerdem ist es (wie im richtigen Leben) an manchen Orten nicht möglich, nachts Informationen zu erlangen, einfach aus dem Grunde, da beispielsweise Shops und Tankstellen um diese Uhrzeit nicht geöffnet haben.

Ein interaktiver Trickfilm

It came from the desert ist ein Storyadventure, das mit hervorragenden Grafiken sowie mit exzellentem Sound glänzen kann. Allein schon das Intro, mit dem das Spiel eingeleitet wird, ist eine Programmierleistung, wie ich sie bei noch keinem Spiel gesehen habe. Aber auch das Motto "Intro klasse, Spiel schlecht" ist hier fehl am Platze. Man hat bei diesem Spiel immer das Gefühl, einen Trickfilm zu sehen, an dem man aktiv teilnehmen kann.

Dabei wurde vom Ablauf des Spieles her Wert auf Realitätsnähe gelegt. So ist es beispielsweise notwendig, sich schlafen zu legen, da sonst (und das ist bei Zweikämpfen mit Rowdies wichtig) die Kondition drastisch abnimmt.

Die Motivation dieses Spieles ist gewiß anfangs recht hoch anzusiedeln, läßt jedoch, nachdem alle Szenarien durchgespielt wurden, drastisch nach. Wer lieber kniffelige Tüftel-Adventures bevorzugt, dem möchte ich dieses Spiel lieber nicht anraten.

(br)

AMIGA DOS Blitzlicht

Name: It came from the desert
Hersteller: Cinemaware
Quelle: Fachhandel
Preis: 99,95 DM

Positiv:
- sehr gute Grafiken
- hervorragender Sound
- Abspeichermöglichkeit

Negativ:
- zur Zeit leider nur in Englisch
- lauffähig nur mit 1 MByte Arbeitsspeicher





Hübsch-häßlich, der X-Out-Oberfiesling im ersten Level

X-OUT

Die Aliens von Alpha Centauri geben sich nicht mit der Annektierung des erdnahen Weltraums zufrieden. Bei denen muß es schon ein schattiges Plätzchen auf dem Meeresgrund sein.

Da sich die ungebetenen Besucher vom fremden Stern gar zu ungebührlich aufführen, wird darüber nachgedacht, wie man die Aliens wieder in den Weltraum zurückbekommt. Projekt X-Out heißt das Zauberwort. Ein schwer bewaffnetes U-Boot, das zu der unterseeischen Basis der Alpha-Centauri-Biester vordringen soll.

Der Spieler, der das X-Out-U-Boot durch das horizontal scrollende Unterwasserszenario steuert, wird feststellen, daß selbst die beste irdische Bewaffnung gegen die Invasoren wenig wirkt. Glücklicherweise kann man in einigen Shops, in die der Spieler jeweils am Ende eines Levels gelangt, Extrawaffen kaufen, die das Leben ungemein erleichtern. Da gibt es vom tollsten Mehrfachschuß über die zielsuchende Rakete bis zum beliebig einsetzbaren Satelliten so ziemlich alles, was für eine heiße Runde Alien-Pulverisieren gebraucht wird. Angsichts der Scharen feindlicher U-Boote, die auf den Spieler einstürmen, bedarf es allerdings außer großer Feuerkraft auch großer Geschicklichkeit, um manche Situation in diesem Unterwasserschocker zu bestehen.

An X-Out kann man seinen Spaß haben. Tolles Intro, gute Grafik, Spitzenmusik und reichlich Action auf dem Bildschirm. Bei den Extrawaffen waren phantasievolle Köpfe am Werk, und insbesondere die frei positionierbaren Satelliten verdienen ein Lob.

(hs)

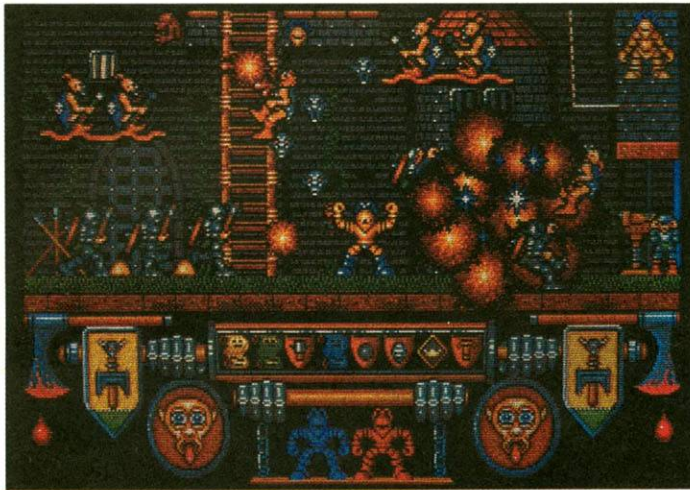
AMIGA DOS Blitzlicht

Name: X-OUT
Hersteller: Rainbow Arts
Quelle: Fachhandel
Preis: 69,95 DM

Positiv:
- dufte Extrawaffen
- gute Musik
- sauberes Scrolling

Negativ:
- bei vielen Objekten auf dem Monitor wird das Spiel deutlich langsamer





Das Königreich von Gargore ist von verfeindeten Clans bevölkert. Nur ein mächtiger Krieger kann den Frieden bringen



Mit Rennbuggys über Stock und Stein

Onslaught

Der Titel dieses Spieles ist schon recht bezeichnend, bedeutet er doch frei übersetzt soviel wie 'Abschlachten'. Es erwartet uns also definitiv kein friedvolles Spiel.

Gargore, so heißt das Land, in dem viele Stämme um die Vorherrschaft kämpfen. Die einzelnen Clans sind schon seit langem untereinander verfeindet, und Fehden werden mit dem Schwert und dem Morgenstern ausgetragen. Da alle Stämme von Gargore durch und durch kriegerisch veranlagt sind, kann es erst Frieden geben, wenn ein Usurpator alle Häuptlinge entmachtet und sich an die Spitze Gargores stellt. Jedes der vielen regionalen Königreiche muß mühsam erobert werden. Zunächst gilt es die Truppen des Feindes zu zerschlagen, dann muß dessen Festung belagert und erobert werden. Ist dies geschafft, gilt es noch den Häuptling niederzuringen, um als Sieger aus dem Feldzug hervorzugehen. Jedes dieser drei Ziele stellt im Spiel einen separaten Abschnitt dar. Die Schlacht und die Belagerung des Feindes finden in einer horizontal scrollenden Landschaft statt, die der Spieler von links nach rechts durchqueren muß, um an den Wappenstab des Feindes zu gelangen. Sind diese beiden Stufen gemeistert, muß der Spieler seine Kräfte mit dem Oberbefehlshaber des Feindes messen. Neben diesen reinen Actionelementen spielt auch ein Hauch Strategie eine Rolle, denn der Spieler kann auf einer Übersichtskarte das Gebiet auswählen, das als nächstes angegriffen werden soll. Actionfreunde werden bei Onslaught voll auf ihre Kosten kommen, auch wenn das Spiel alles andere als friedlich ist.

(hs)

AMIGA DOS Blitzlicht

Name: Onslaught
Hersteller: Hewson
Quelle: Fachhandel
Preis: 84,95 DM

Positiv:
- actionreiches Spielgeschehen
- gute Grafik
- guter Sound

Negativ:
- stellenweise brutale Hintergrundgrafik



Power Drift

Ein Rennbuggy, harte Rennstrecken und verwegene Fahrer, – das sind die Ingredienzen, aus denen Power Drift gemixt wurde. Gute Nerven und einen bleiernen Gasfuß müssen Sie mitbringen.

Vier Runden lang gilt es, sich gegen eine große Konkurrenz durchzusetzen, um als einer der ersten Fahrer über die Ziellinie zu gehen. Zwölf Piloten gehen bei jedem Power-Drift-Rennen an den Start der verschlungenen Rennstrecke. Nach dem Startschuß versucht natürlich jeder Fahrer sogleich, sich an die Spitze des Feldes zu setzen. Dem Spieler wird seine Position im Fahrerpulk durch eine große, gelbe Ziffer angezeigt. Neben den scharfen Kurven weisen die Rennstrecken als kleines Extrahindernis Brücken auf, die das Rennen zusätzlich erschweren, da sich hinter einer nicht einsehbaren Steigung durchaus eine scharfe Kurve befinden kann. Konnte der Spieler sich auf einem der ersten drei Ränge plazieren, geht es im nächsten Rennen auf einer neuen Strecke, die ein bißchen schwieriger ist, weiter.

Power Drift ist eines jener Spiele, die in der Gefolgschaft eines Hitspiels auf den Markt kommen und sich daran orientieren. So sieht man Power Drift unschwer an, daß da irgendwo Outrun Pate gestanden hat. Aber während Outrun für den Amiga ganz schön schlecht war, gibt Power Drift spielerisch etwas her und macht zumindest Spaß. Aber leider ist Power Drift im Grunde nur ein einfaches Rennspiel, auch wenn es hier Brücken gibt, über die hinweg und darunter durch gefahren werden muß. Meine ehrliche Meinung: Ein ganz nettes Rennspiel, das nur wenig Neues präsentiert. Haben wir alles schon einmal besser gesehen.

(hs)

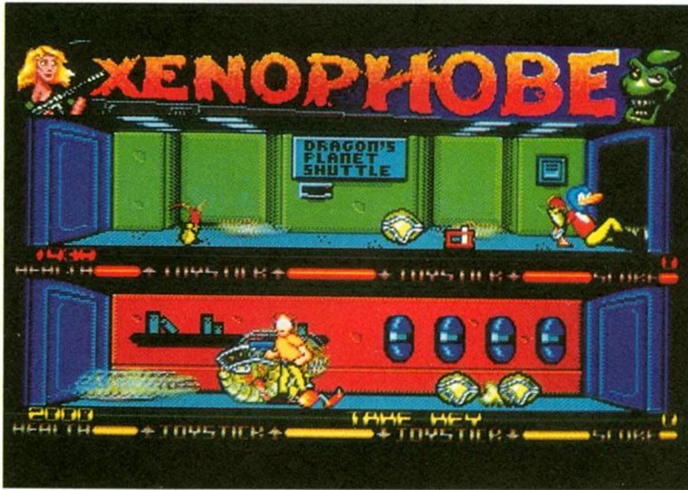
AMIGA DOS Blitzlicht

Name: Power Drift
Hersteller: Activision
Quelle: Fachhandel
Preis: 84,95 DM

Positiv:
- passables Rennspiel

Negativ:
- ruckelige Animation
- wird bald langweilig





Der kosmische Kammerjäger ist da – mit Xenophobe auf Alienjagd

Xenophobe

Die grünen, schleimigen Aliens aus dem hintersten Winkel der Galaxis haben einen neuen Job: Sie überfallen Raumstationen und fallen über deren Besatzungen her.

Aber so, wie hier der Kammerjäger anrückt, wenn es gilt, Ungeziefer zu beseitigen, gibt es auch für das Eliminieren von Aliens Spezialisten. Xenophobe versetzt Sie ohne Umschweife in die Rolle eines solchen Alien-Kammerjägers, der infizierte Raumstationen und Planeten säubern muß.

Um die Xenomorphen auszumerzen, müssen Sie an Bord des infizierten Schiffes gehen, um dort, mit einem Laser in der Hand, die grünen Schleimlinge einen nach dem anderen in ebenso grüne Pfützen zu verwandeln. Leider ist eine der ersten Aktionen, wenn die Alieninvasion offenkundig geworden ist, die Aktivierung der automatischen Selbstvernichtungsanlage. Diese zählt natürlich gnadenlos den Countdown bis zur Vernichtung herunter, so daß dem Jäger nur eine begrenzte Zeitspanne bleibt, um seine Aufgabe zu erfüllen. Dazu muß er durch die Korridore der Stationen wandern und alles, was wie ein Alien aussieht, ausmerzen. Unterwegs können Wertgegenstände, Bomben und andere Waffen, die herumliegen, aufgesammelt werden.

Wollen zwei Spieler auf Alienjagd gehen, wird der Bildschirm gesplittet. Zwar geht es in Xenophobe rasant zur Sache, und es darf geballert werden, was das Zeug hält, jedoch ändert das nichts an der Tatsache, daß auch dies auf die Dauer langweilig wird. Der einzige echte Pluspunkt dieses Spiels ist seine Musik. Die hat echten 'Drive' und läßt manch anderen Schwachpunkt vergessen.

(hs)

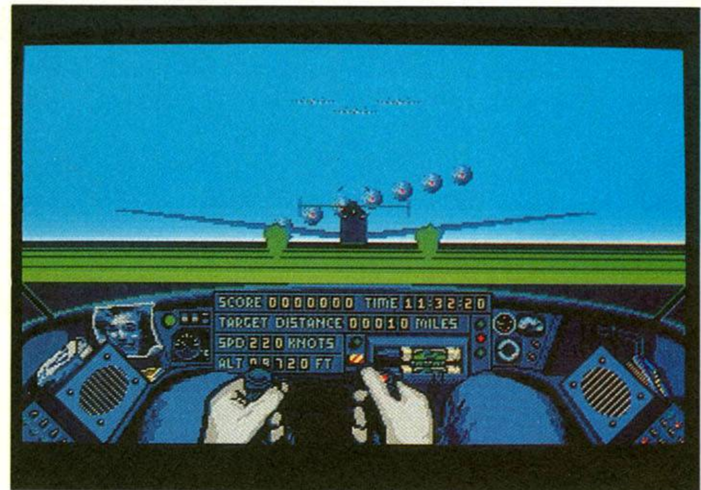
AMIGA DOS Blitzlicht

Name: Xenophobe
Hersteller: Micro Style
Quelle: Fachhandel
Preis: 89,– DM

Positiv:
- reichlich Action auf dem Bildschirm
- gute Musik

Negativ:
- wenig abwechslungsreich





Erst mit dem MG gegen feindliche Jäger und dann zum Bombenabwurf – Lancaster ist alles andere als friedfertig

Lancaster

Mal wieder Lust auf ein richtiges Stück kriegerischer Actionsoftware, das am besten mit abgeschalteter Moral gespielt wird? Dann könnte Lancaster vielleicht das richtige für Sie sein.

Die Lancaster-Maschine war der leistungsfähigste Bomber, den die englische Luftwaffe während des zweiten Weltkrieges gegen Deutschland ins Feld schickte. Dieser Bomber war mit schwenkbaren MG-Stationen ausgestattet, so daß feindliche Jäger von allen Seiten eine wehrhafte, fliegende Festung vorfanden. Lancaster versetzt Sie in die Rolle eines Heck-MG-Schützen an Bord eines Lancasterbombers, der gerade auf Feindflug unterwegs ist.

Der Spieler bekommt zu Beginn des Spiels auf einer Landkarte ein Zielgebiet vorgegeben, das erreicht werden muß. Der Flug dorthin wird natürlich immer wieder durch Jägerangriffe behindert.

Taucht eine feindliche Jägerstaffel auf, blendet das Spiel in den Heckausblick um. Der Spieler kann nun mittels eines zu einem Fadenkreuz umfunktionierten Mauszeigers ballern, was das Zeug hält.

Die anfliegenden feindlichen Maschinen und auch alle anderen Flugszenen werden in gefüllter Vektorgrafik dargestellt. Kommen allerdings einmal mehr als drei Flugzeuge gleichzeitig, wird die Animation deutlich langsamer.

Wurde der Zielort wohlbehalten erreicht, darf der Spieler feindliche Bodenanlagen bombardieren.

Lancaster ist nicht Fleisch, nicht Fisch. Keine richtige Simulation, kein richtiges Actionspiel. Was bleibt, sind ein paar mäßig programmierte Vektoranimationen, die niemanden vom Sessel reißen.

(hs)

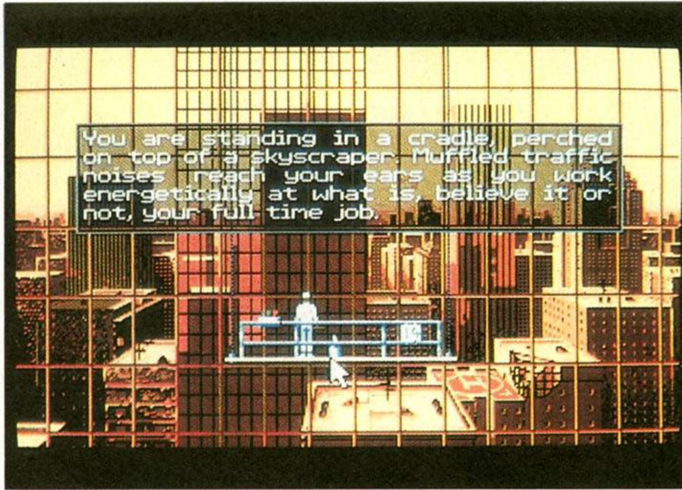
AMIGA DOS Blitzlicht

Name: Lancaster
Hersteller: Actual Screenshot
Quelle: Fachhandel
Preis: 69,95

Positiv:
- Status speicherbar

Negativ:
- Kriegsspiel reinsten Wassers
- mäßige Vektorgrafik
- magere Soundeffekte





Unser Held ist ein einfacher Fensterputzer, der erst noch zeigen muß, was in ihm steckt

FUTURE WARS

(Time Travellers)

Die Erde ist der letzte Planet der Menschen, die sich im Krieg mit den Crughonen befinden. Sie wird lediglich durch einen Schutzschirm vor den feindlichen Angriffen bewahrt. Eines Tages brechen zwei der Aliens zur Erde durch, benutzen die Chronoporter und beamen sich ins Jahr 1304...

So beginnt Future Wars, ein interaktives Adventure, das den Untertitel Time Travellers, Zeitreisende, trägt. Zu Beginn des Spiels haben Sie noch keine Ahnung von diesen zukünftigen Ereignissen, und vor allem können Sie sich nicht vorstellen, daß Sie selbst in die Ereignisse verstrickt und dabei so manches Abenteuer erleben werden.

Sie als unser Held, wir wollen ihn Case nennen, sind ein einfacher Fensterputzer in einem Hochhaus. Case fährt mit einem Aufzug an den Fenstern entlang, die er putzt. Ausgerechnet als Ed, sein Chef, zuschaut, stößt er mit dem Fuß den Putzeimer um. Darüber gerät Ed, der sowieso schon nicht so gut auf Case zu sprechen ist, sehr in Rage. Nach der gehörigen Standpauke schließt Ed sein Fenster nicht richtig. Da die Lust auf Arbeit verflogen ist, nimmt Case den Eimer und fährt zu dem halboffenen Fenster.

Die Gelegenheit ergreifend, betritt er ein kleines Vorzimmer, dem ein Bad angegliedert ist. Eine gründliche Untersuchung des Raumes fördert einige nützliche Dinge zu Tage. So findet er unter anderem unter dem Teppich einen kleinen Schlüssel.

Ein Adventure, bei dem man Ideen haben muß

Als kleinen Scherz füllt Case seinen Eimer wieder mit Wasser und plaziert ihn über der Tür, hinter der er seinen Chef vermutet. Anschließend verläßt er das Zimmer durch eine weitere Türe. Beim Öffnen dieser Türe macht er einen solchen Lärm, daß sein Chef seine Tür aufreißt und der Eimer genau auf seinen Schultern landet, Case macht sich aus dem Staub ins nächste Zimmer, bevor sein Chef merkt, wem er diese unfreiwillige Dusche zu verdanken hat. In diesem Raum findet er eine große militärische Landkarte und eine Schreibmaschine, auf deren Carbonfarbband eine Nummer zu lesen ist. Wofür das gut ist und vor allem was die bösen Crughonen damit zu tun haben, wird man wohl erst im weiteren Spielverlauf erfahren.



Ob sein Chef ihm den Scherz mit dem Eimer wohl übelnehmen wird?

Future Wars ist ein neues, interessantes Adventure, das eigentlich nur zwei wesentliche Nachteile besitzt: Zum einen ist es zu schwer, so daß bei einem Adventure-Anfänger leicht Ärger auftreten kann und dieser es dann in die Ecke legt. Zum anderen muß man seine Figur für banale Dinge pixelgenau auf dem Bildschirm plazieren. Dies ist mit einigen Schwierigkeiten verbunden, da man die Figur dreidimensional durch die Räume bewegen kann, der Mauszeiger aber nur zweidimensional bewegbar ist. Auch hier entsteht leicht Frust.

Das Spiel ist komplett per Maus spielbar. Die möglichen Befehle erhält man nach einem Druck auf die rechte Maustaste in einem Fenster.

Die Räume werden in Form von Fenstern auf dem Bildschirm dargestellt, deren Größe sich mit der Größe der Räume verändert. Sie schwankt dabei zwischen sehr klein bis hin zu bildschirmfüllend.

Findet man einen Gegenstand, so wird er vergrößert dargestellt, und man kann ihn genauer untersuchen. Auch die Textfenster verdienen besondere Beachtung: Sie werden durchscheinend dargestellt, so daß man alle Details, die sich hinter den Fenstern befinden, noch erkennen kann. Kurz und bündig: Future Wars, Time Travellers ist ein Adventure, das sich allein wegen seiner Grafik und dem Sound zu spielen lohnt, wenn es auch etwas schwer anmutet.

(Robert Marz/hs)

AMIGA DOS Blitzlicht

Name: Future Wars Time Travellers

Hersteller: Delphine Software

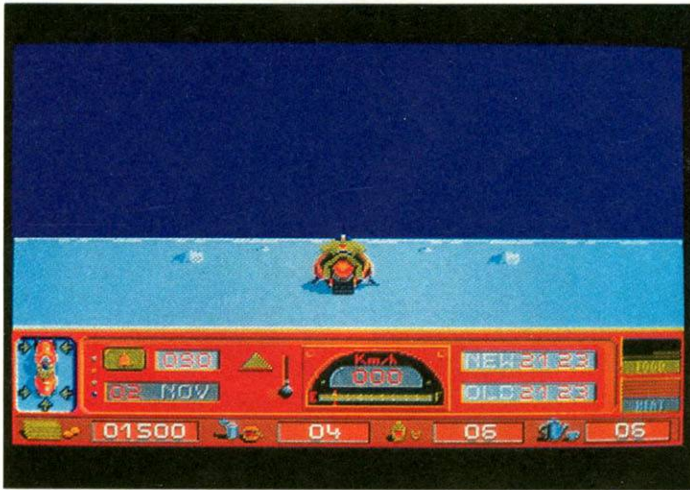
Quelle: Fachhandel

Preis: stand bei Redaktionsschluß noch nicht fest

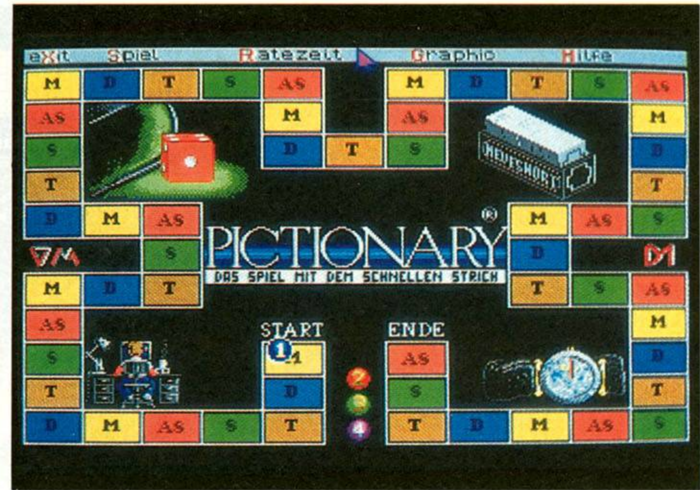
Positiv:
- gutes Adventure mit gutem Sound und guter Grafik
- alle Befehle über Maus erreichbar

Negativ:
- sehr hoher Schwierigkeitsgrad
- pixelgenaue Plazierungen für einfache Handlungen nötig





Mit dem Schneescooter auf dem Weg zu den Wettrennen der Weltmeisterschaft. Unterwegs können sie auch jagen.



Beim Malen und Erraten der Begriffe kommt sehr schnell Stimmung auf

Skidoo

Hatten Sie schon einmal Lust, mit einem Schneescooter durch die endlosen, schneebedeckten Weiten Nord-Kanadas zu fahren?

In diesem Spiel haben Sie es sich zum Ziel gemacht, die diesjährigen Weltmeisterschaften im Scooterfahren zu gewinnen. Die Qualifikationsrennen finden in verschiedenen Metropolen Kanadas statt, und wer am Ende der sechsmonatigen Saison die meisten Punkte gesammelt hat, ist Weltmeister. In jeder Stadt werden während der Saison zehn Rennen veranstaltet. Um an möglichst vielen teilzunehmen, müssen Sie also zwischen den Städten hin- und herfahren. Dabei hilft eine Karte, die dem Spiel beiliegt. Unterwegs kann viel passieren: die Vorräte können ausgehen, oder der Scooter kann durch unvorsichtige Fahrweise Schaden nehmen, was bei einer Spitzengeschwindigkeit von 186 km/h sehr schnell geschehen kann. Es gibt für den Notfall glücklicherweise Abschleppdienste, die den Scooter reparieren und an der gleichen Stelle wieder absetzen. Sollten Geld oder die Vorräte ausgehen, so kann man während der Fahrt auch Wölfe und Karibus jagen. Wolfsfelle können dann verkauft werden, und Karibufleisch ist ausgesprochen wohlschmeckend. Skidoo hat sicherlich seine spielerischen Reize und wird wohl unter eingefleischten Rennspielfans auch Freunde finden. Jedoch gestaltet sich der Spielablauf recht zäh und nur wenig abwechslungsreich. Die Fahrt durch die Weiten Kanadas ist bis zu einem gewissen Grad realistisch gestaltet. Allerdings betrifft dies primär die Eintönigkeit der Fahrt. Die atemberaubende Landschaft von Skidoo kann bestenfalls eine halbe Stunde überzeugen.

(Robert Marz/hs)

AMIGA DOS Blitzlicht

Name: Skidoo
Hersteller: Tomahawk
Quelle: Fachhandel
Preis: 64,95 DM

Positiv:
- speicherbarer Spielstand
- gute Grafik

Negativ:
- wenig abwechslungsreicher Spielverlauf
- Streckenbegrenzungen nicht groß genug



Pictionary

Einen Begriff zeichnerisch darstellen und von anderen erraten lassen, ist das Grundprinzip mancher Quizsendungen und des Computerspiels Pictionary. Dabei werden die Begriffe mit einem integrierten Malprogramm gezeichnet.

Eigentlich basiert Pictionary auf einem erfolgreichen Brettspiel. Und dementsprechend kann das Pictionary Computerspiel mit einer ganzen Gruppe von Teilnehmern gespielt werden.

Der Amiga verwaltet das Spielgeschehen. Er mischt sich in die eigentliche Handlung nur wenig ein und ist lediglich Verwalter. Er zeichnet das Spielfeld, würfelt und zählt die Punkte der Spieler. Lediglich wenn Einzelpersonen spielen, zeichnet er höchstpersönlich die gesuchten Begriffe.

Der Computer gibt auch an, welche Begriffe gezeichnet werden sollen. Der Zeichner kann dann den Begriff von einer der dem Spiel beiliegenden Karten ablesen und ihn mittels eines im Spiel enthaltenen Malprogramms zeichnen. Falls die Rater glauben, den Begriff erraten zu haben, können sie den Malvorgang jederzeit durch Drücken einer Taste unterbrechen. Wenn sie richtig lagen, so wird die übergebliebene Zeit gutgeschrieben. Ob richtig geraten wurde, machen die Spieler unter sich aus, der Computer fragt lediglich, ob oder ob nicht. Bei richtiger Antwort darf noch einmal gewürfelt werden. Gewonnen hat das Team, das als erstes das letzte Feld erreicht hat und den letzten Begriff richtig errät. Insbesondere in Gesellschaft bringt Pictionary viel Spaß und ist dazu angetan, in die Fußstapfen von Trivial Pursuit zu treten.

(Robert Marz/hs)

AMIGA DOS Blitzlicht

Name: Pictionary
Hersteller: Domark
Quelle: Bomico/Fachhandel
Preis: 89,95 DM

Positiv:
- Malprogramm enthalten
- Bilder im IFF-Format speicherbar
- durch viele verschiedene Begriffskarten große Abwechslung
- bis zu vier Mannschaften

Negativ:
- Ratezeit zu kurz
- Spielstand nicht speicherbar





Diamantenjagd im Geisterschloß, da sind Timing und Sprungkraft gefragt

Danger Castle

Eine unheimliche Burg voller Kobolde, wandelnder Ritterrüstungen und kleiner roter Teufelchen ist der Schauplatz einer fröhlichen Diamantenjagd, die den Spieler durch die Gemächer des Gemäuers führt.

Die großen Zimmer sind in viele, durch Abgründe durchbrochene Ebenen unterteilt, die der Spieler mit seiner Figur, einem kleinen braunhaarigen Jungen, erkunden muß. Dies tut der Kleine des schnöden Mammons willen, denn er macht seinen Weg durch die Burg nur, um die überall versteckten Juwelen einzusammeln. Dabei muß er aber nicht nur Geschick beim Laufen und Springen an den Tag legen, sondern auch noch mit vielen verschlossenen Türen und wenigen Schlüsseln herumhantieren. Ein Unterfangen, das durch die umherirrenden Spukgestalten, die dem Burschen bei jeder Berührung Energie abziehen, nicht gerade erleichtert wird. Zum Glück erscheinen dann und wann kleine magische Gegenstände, die dem Kleinen Hilfe verheißen. Mal ergänzen sie das Punktekonto, mal lähmen Sie die Monster für einen Moment. Jedesmal wenn alle Schätze eines Gewölbes geplündert wurden, gelangt der Spieler einen Raum weiter. Ein magischer Schlüssel, der von Zeit zu Zeit erscheint, ist ein echtes Kleinod, das es zu ergattern gilt.

Danger Castle ist ein nettes Spiel, das einen niedlichen Eindruck macht und Spaß macht. Grafik und Soundeffekte brechen zwar keine Rekorde, sind aber durchaus sehens- und hörenswert. Größtes Minus sind die zwanzig Level, die sich nach wenigen Spielen als eindeutig zu wenig erwiesen, schade.

(hs)

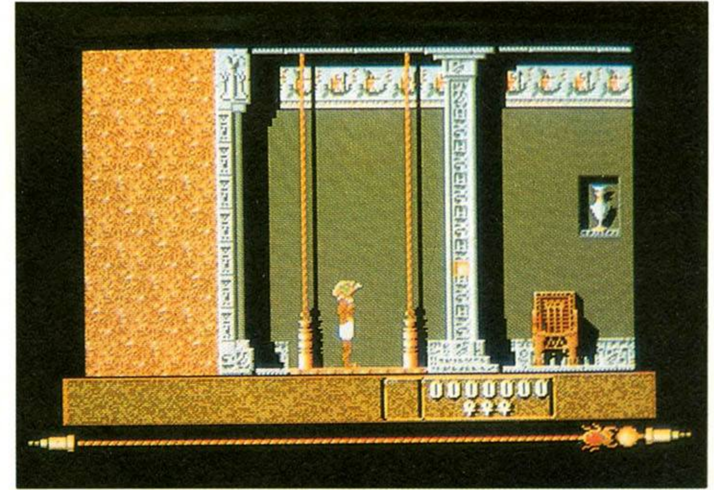
AMIGA DOS Blitzlicht

Name: Danger Castle
Hersteller: Kunert Soft
Quelle: Fachhandel
Preis: 39,- DM

Positiv:
- niedliche Grafik

Negativ:
- zu wenig Levels





Auf der Suche nach den Teilen seines Vaters wird Horus immer wieder von fliegenden Hieroglyphen angeriffen

Eye of Horus

Horus, Sohn des Gottes Osiris, muß seinem Vater einen letzten Dienst erweisen: Er soll seinen bösen Onkel Seth ein zweites Mal besiegen. Bevor Osiris ihm die Kraft dazu verleihen konnte, zerteilte Seth Osiris Körper in sieben Stücke, die er in einem Grab versteckte.

Horus muß nun zunächst einmal die Teile von Osiris' Körper wieder zusammenfügen. Dazu bewegt er sich, von Ihnen, dem Spieler, gesteuert, durch das Labyrinth der Grabkammern. Diese sind durch mystische Fahrstühle miteinander verbunden. Da Horus ein Halbgott ist, besitzt er eine besondere Eigenschaft: Er kann sich nach Belieben in einen Falken verwandeln und entsprechend schnell durch die Luft fliegen. Bis jetzt hört sich die ganze Sache ja noch nicht sehr schwer an, werden Sie sich denken, was kann in einem Grab denn schon Großartiges passieren? Im Prinzip haben Sie recht. Seth will allerdings nicht zusehen, wie Horus sich darauf vorbereitet, ihn zu vernichten. Er beeinflußt die Hieroglyphen an den Wänden in der Art, daß sie lebendig werden, durch die Gegend fliegen und Sie attackieren. Horus aber kann sich wehren: Er verschießt Kügelchen und aktiviert gefundene magische Amulette, von denen es eine Menge gibt.
Mit den Amuletten kann er sich zum Beispiel eine Karte anzeigen lassen oder weitere Götter um Hilfe anflehen. Eye of Horus gestaltet sich als Action-Adventure der gehobenen Mittelklasse mit mäßigen Grafiken und Sound. Wenn die Spielidee an sich auch recht gut ist, denn die ägyptische Mythologie gibt viel her, so hat man das Spiel doch bald über, da nichts wesentlich Neues passiert.

(Robert Marz/hs)

AMIGA DOS Blitzlicht

Name: Eye of Horus
Hersteller: Logotron
Preis: 84,95 DM
Quelle: Fachhandel

Positiv:
- viele Extras in Form von Amuletten
- gute Steuerung

Negativ:
- Spielverlauf eintönig





Klar, sauber und ohne Schnörkel, Pinball Magic ist ein gradliniges Flippervergnügen



Drivin Force – auf Teufel komm raus über Stock und Stein

Demnächst auf Ihrem Computer

Sportspaß, Alienattacken und Flipperfreuden erwarten den spielbegeisterten Amigafreund. Nach wie vor geht der Trend zu technisch ausgefeilten Programmen, die es aber nicht versäumen, auch die spielerische Seite mit guten Ideen abzudecken. Mal schnell ein paar Sprites über den Bildschirm hüpfen lassen, das war einmal. Heute braucht selbst ein Flipper mehrere verschiedene Spielfelder.

Pinball Magic

Alle diejenigen unter Ihnen, die sich nostalgisch die guten alten Flipper wieder wünschen, die noch durch und durch mechanisch waren, sollten sich Pinball Magic einmal genauer ansehen. Da kommt beim Spielen richtig nostalgisches Flipperfeeling auf. Denn hier wurde kein High-Tech-Flipper mit digitaler Punkteanzeige als Vorbild verwendet, sondern tatsächlich ein uraltes Gerät, das noch mit Zählrollen arbeitet. Grafisch und soundtechnisch vielversprechend umgesetzt, wird Pinball Magic die Herzen aller Flipper Freunde höher schlagen lassen.

Drivin Force

Autofahrten über Stock und Stein, gleich gegen eine ganze Horde anderer Fahrer, und das Ganze auch noch als Wettkampf, das sollte alle Fans von Rennspielen hellhörig gemacht haben. Mit Drivin Force steht uns ein neuer Vertreter des Genres ins Haus, der einen recht guten Eindruck macht. Was auf der Demo vom Sound und der Geschwindigeit des Spielablaufs zu sehen und zu hören ist, läßt ein rasantes Rennspiel erwarten, bei dem nicht nur die Optik stimmt.

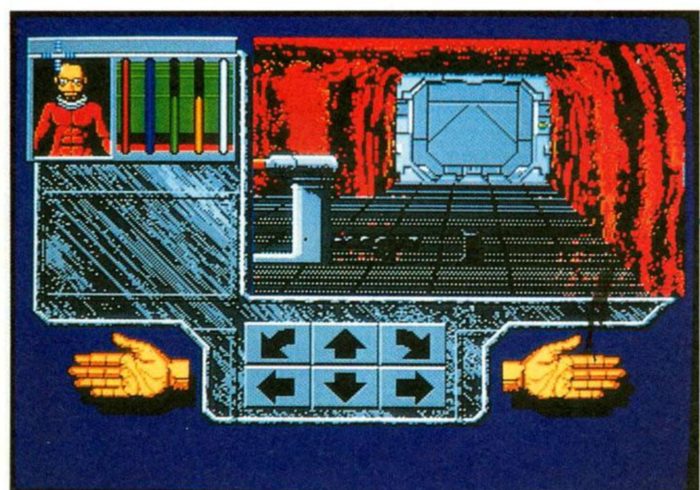
Infestation

Psygnosis, die 16-Bit-Magier aus England, schicken sich an, geneigte Computerspieler auf einen einsamen, unwegsamen Mond auszusetzen. Sie müssen diesen Trabanten und alle darauf befindlichen Baulichkeiten untersuchen. Neben einer gehörigen Portion Action erwartet den Spieler ein ausgeklügeltes Abenteuer und viele Stunden aufregender Erkundung in einem sauber programmierten 3-D-Szenario, das mit gefüllten Vectormodellen arbeitet.

Xenomorph

Dieses Spiel führt Sie zwar auch auf eine Raumstation, doch hier erwartet Sie eine Begegnung ganz anderer Natur. Aliens der scheußlichsten Sorte haben die Station überfallen und alle Besatzungsmitglieder gefangengenommen. Bei diesem Spiel treffen sich Action, Abenteuer und Rollenspiel. Die Programmierer haben versucht, das Beste aus diesen drei Genres zu einem neuen Spiel zusammenzufügen. Wie gut dieser Versuch gelungen ist, werden wir sehen. (hs)


Infestation – erkunden Sie die unwirtlichen Weiten eines einsamen Mondes


Alien und Raumstation, in Xenomorph wieder glücklich vereint



Strategie- und Rollenspiel geben sich ein Stelldichein, Legend lädt zum Erobern ein

Legend

Die Legende beginnt, als der erste große Krieg sein zweites Jahrtausend erreichte. Es war ein Krieg zwischen Gut und Böse, in dem keine Seite den Sieg erringen konnte.

Ein einzelner Mann befriedete diesen Krieg mittels der Macht seines Zauberstabes. Nach seinem Tode entbrennt unter den Erben der Streit um den Stab, der dabei zerbricht. Dadurch flammt der Krieg erneut auf. Nun stehen sich die Erben gegenüber. Von nun an liegt es an Ihnen, über das weitere Schicksal dieser Welt zu entscheiden. Sie schlüpfen in die Rolle eines der beiden Kriegsherren.

In schönster Strategiespielmanier gilt es zuerst, möglichst viele Länder an sich zu reißen, um durch die eingetriebenen Steuern Armeen und Ausrüstungsgegenstände zu kaufen. Hat man genügend Geld akquiriert, kann man Generäle rekrutieren, die Truppen befehligen. Oder man kann Diebe, Drachen und Meuchelmörder engagieren, die dem Gegner hinterrücks schaden, oder aber einfach weitere Länder erobern.

Eine weitere Möglichkeit, seiner Feinde Herr zu werden, sind Zaubersprüche. Mit diesen kann man, je nach Kraft, dem Feind zusetzen. Ähnlich aufgebaut wie Defender of the Crown, bietet Legend doch eine ganze Masse neuer Features, was wohl der etwas ungewöhnliche Handlungsort mit sich bringt. Aber auch die vielen Extras können nicht über die Langeweile hinwegtäuschen, die sich nach einiger Spielzeit einstellt. Hinzu kommt, daß sowohl Grafik als auch der Sound nicht zu Begeisterungsstürmen verleiten können, so daß Legend nur bedingt den absoluten Fans dieses Genres empfohlen werden kann.

(hs)

AMIGA DOS Blitzlicht

Name: Legend
Hersteller: Actual Screenshots
Quelle: Fachhandel
Preis: 69,95 DM

Positiv:
- Zwei-Spieler-Option

Negativ:
- bescheidene Grafik
- schnell langweilig
- kein Sound



Impressum

Impressum

Herausgeber
Christian Widuch

Chefredaktion
Markus Matejka (mm)

Redaktion
Jürgen Borngießer (jb), Bernhard Rinke (br), Heinrich Stiller (hs), Joachim Freiburg (jf), Martin Schlöter (ms), Vera Brinkmann (vb)

Freie Autoren dieser Ausgabe
Edgar Meyzis, Bernd Rudolf, Robert Marz, Jürgen Seibel, Garry Glendown, Ulf Peterson, Michael Trebing, Michael Anton, Günter Noppeney, Michael Cordes, Niklas Nebel

Redaktionsassistenz
Anke Kerstan (ke), Susanne Eska (es)

Koordination
Stefan Ritter

Produktionsleitung
Gerd Köberich

Bereichsleitung
Claudia Ebbrecht (Fotosatz/Lektorat), Margarete Schenk, Helmut Skoupy (Montage/Reprografie)

Layout
Yvonne Hendricks

Fotografie
Christian Heckmann, Klaus Jatho

Fotosatz
Gabriela Joseph

Lektorat
Susanne Lessinger

Montage/Reprografie
Manuela Eska, Dieter Schnobl

Werbegestaltung
Mohamed Hawa, Petra Küch

Anzeigenverkaufsleitung
Wolfgang Schnell

Anzeigenverkauf
DMV-Verlagsbüro München
Zaunkönigweg 2c, 8000 München 82
Telefon (089) 4391087, Telefax 089/4391080
Leitung: Britta Fiebig
Anzeigenverkauf: Monika Schöbel, Jens Dhein, Peter Schätzle

Anzeigenpreise
Es gilt die Anzeigenpreisliste Nr. 1 vom 01.01.1990

Anschrift Verlag/Redaktion:
DMV Daten & Medien-Verlag
Widuch GmbH & Co. KG
Fuldaer Straße 6
3440 Eschwege, Telefon (0 56 51) 809-0,
Telefax (0 56 51) 809-333

Vertrieb
Verlagsunion Erich Pabel-Arthur Moewig KG (VPM), Friedrich-Bergius-Straße 20, 6200 Wiesbaden

Druck
Druckerei Jungfer, 3420 Herzberg

Bezugspreise
»AMIGA DOS« erscheint monatlich.
Einzelpreis DM 6,50/sfr. 6,50/öS 52,–

Abonnementpreise
Die Preise verstehen sich grundsätzlich einschließlich Porto und Verpackung.

Inland:
12 Ausgaben: DM 70,–
6 Ausgaben: DM 35,–
Europäisches Ausland:
12 Ausgaben: DM 100,–
6 Ausgaben: DM 50,–
Außereuropäisches Ausland:
12 Ausgaben: DM 120,–
6 Ausgaben: DM 60,–

Bankverbindungen:
Postscheck Frankfurt/M: Kto.-Nr.: 23043-608
Raiffeisenbank Eschwege:
BLZ: 522 603 85, Kto.-Nr.: 245 7008

Die Abonnementbestellung kann innerhalb einer Woche nach Auftrag beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich widerrufen werden. Zur Wahrung der Frist reicht der Poststempel. Das Abonnement verlängert sich automatisch um 6 bzw. 12 Monate, wenn es nicht mindestens 6 Wochen vor Ablauf beim Verlag schriftlich gekündigt wird.

Der Schweiz-Auflage liegt ein Prospekt des WEKA-Verlages bei

Die Inserenten

Diskettenlaufwerke

3 1/2" Laufwerk AMIGA 2000 intern komplett mit Einbaukit und Anleitung	DM 159,–
3 1/2" Laufwerk für alle AMIGAs extern abschaltbar, Busdurchführung, AMIGAfarben	DM 229,–
5 1/4" Laufwerk für alle AMIGAs extern abschaltbar, Busdurchführung, 40/80 Tracks, AMIGAfarben	DM 279,–

Disketten

3 1/2" No Name 2DD	10 St.	DM 16,95
3 1/2" Verbatim Verex 2DD	10 St.	DM 25,–
3 1/2" Sentinel COLOR-Disk lieferbar in Grün, Gelb, Rot, Orange und Weiß	10 St.	DM 29,90
5 1/4" No Name 2S2D	10 St.	DM 5,90
5 1/4" Verbatim Verex	10 St.	DM 14,90

Staffelpreise auf Anfrage

Festplatten

Kapazität	Speed	Filecard	A2000	A500
20MB/3 1/2"	35ms	898,–	848,–	1048,–
30MB/5 1/4"	65ms	-	898,–	1098,–
30MB/3 1/2"	35ms	1148,–	1098,–	1298,–
40MB/5 1/4"	28ms	-	1198,–	1398,–
50MB/3 1/2"	35ms	1398,–	1348,–	1548,–
60MB/5 1/4"	28ms	-	1398,–	1598,–

Alle unsere Festplatten werden wahlweise mit ALF V1.6 ausgeliefert.
ALF V2.0 auf Anfrage.

AUTOBOOTMODUL für Amiga

Amiga 2000 Autobootmodul	DM 119,–
Amiga 500 Autobootmodul	DM 149,–

(Bitte Festplatten- und Controllertyp angeben)

Festplatten-Interface — DM 99,–
Die Adapterplatine paßt den PC-BUS eines Festplatten-Controllers an den AMIGA-BUS an. (Bitte Rechnertyp angeben)

ALF 1.6 Festplattentreiber	DM 98,–
ALF 2.0 Festplattentreiber	DM 198,–
MFM-Set OMTI 5520B, ALF V1.6, Festplatten-Interface, Kabelsatz	DM 349,–
RLL-Set OMTI 5527B, ALF V1.6, Festplatten-Interface, Kabelsatz	DM 369,–

AMIGA-Computer

Amiga 2000	DM 1898,–
Amiga 500	DM 928,–
Amiga 500 BTX	DM 998,–
Harddisk A590 20MB für A500	DM 998,–
Colormonitor Commodore 1084P	DM 598,–

RAM-Erweiterung AMIGA 500 auf 1MB — DM 189,–

mit Uhr, abschaltbar
dto. Platine mit Uhr & Schalter ohne RAMs — DM 79,–
RESTPOSTEN, Lieferung nur solange der Vorrat reicht

RAM-Erweiterung II AMIGA 500 auf 1MB — DM 199,–

Megabittechnologie, mit Uhr, abschaltbar
dto. Platine mit Uhr & Schalter ohne RAMs — DM 79,–

RAM-Erweiterung AMIGA 500 auf 2,3MB — DM 698,–

mit Uhr, abschaltbar, intern!
dto. Platine mit Uhr & Schalter ohne RAMs — DM 198,–

RAM-Erweiterung AMIGA 2000 — DM 848,–

8MB mit 2MB bestückt, autokonfigurierend, einfach durch zusätzliche RAMs und Jumper umstecken aufrüstbar, keine neuen PALs erforderlich.
dto. Platine teilbestückt ohne RAMs — DM 498,–
RAM-Satz für 2 Megabyte — DM 498,–

AMIGA-Bremse * der Highscore-Killer * — DM 39,50

- regelt die Geschwindigkeit stufenlos bis zum Stillstand
- ideal für schnelle Games und Bildschirmfotografie

BOOT-Selector für Amiga — DM 14,50

wahlweise Booten von DF0: oder DF1: oder DF2: oder DF3:
bei Bestellung bitte angeben

Drive-Expander — DM 39,–

- für externe Laufwerke ohne Busdurchführung, einstellbare Laufwerksnummer, keine Kabellängenprobleme, abschaltbar
  bei Verwendung eines beliebigen Boot-Selectors, kann von jedem Laufwerk gebootet werden

Maus & Joystick-Adapter — DM 44,50

- für gleichzeitigen Anschluß von Maus und Joystick, mit LED-Anzeige, alle Maussteuerleitungen elektronisch geschaltet

Modem Set Discovery 2400C — DM 448,–

- inkl. AMIGA-Anschlußkabel und DFÜ-Software

Das Modem ist ein Exportmodell ohne Postzulassung! Inbetriebnahme strafbar

BTX/VTX Decoder mit FTZ-Zulassung — DM 248,–

Trackdisplay A2000 intern DF0: & DF1: — DM 98,–

(neuer Power-LED Träger mit integrierten 7-Segmentanzeigen)

Trackdisplay extern DF0: bis DF3: — DM 79,–

für jedes Laufwerk einstellbar

Drucker, Farbbänder, Kabel, Staubschutzhauben und Software auf Anfrage

»PowerFire« Das Superding! — DM 29,50

Dauerfeuerinterface für Joystick und Maus
- optimale Impulsfolge für jedes Game einstellbar
- Dauerfeuer wird über Feuer- bzw. Maustaste aktiviert
- einfach zwischen Maus/Joystick und Rechner stecken
- abschaltbar

Kick-ROM — DM 49,–

- Kickstartumschaltplatine für zwei OriginalROMs

OriginalROM 1.2 oder 1.3	DM 65,–
Kick-ROM mit einem OriginalROM	DM 98,–
KickstartEproms 1.2/1.3/Guardian	DM 99,–

Kickstartumschaltplatine 3-fach — DM 59,–

- für zwei OriginalROMs und eine Epromversion

Umschaltplatine mit einem Epromsatz	DM 155,–
Umschaltplatine mit einem OriginalROM	DM 108,–

Midi-Interface — DM 89,–

In/Thru/2*Out

Genlock-Interface — DM 248,–

nur Computer/nur Video/beide Bilder gemischt, durchgeführter Monitorport, externes Gehäuse

Professional View

Video-Digitizer der neuen Generation — DM 998,–
On Board RGB-Splitter bereits integriert !!
On Board Genlock-Interface bereits integriert !!
Realtime-Digitalisierung in 16 Graustufen, 25 Bilder/s
HAM-Modus mit 4096 Farben
kurze Realtime Animationen in s/w, Übertragung 3,5MB/s !
Alle Bilder und Animationen auf Diskette speicherbar
für Animationen mind. 1MB RAM
unterstützt 1MB ChipRAM

Software

X-Copy II	DM 49,–
X-Copy II mit Hardwarezusatz	DM 69,–
Sherlock V2.0 AntiVirusTools	DM 49,–
Turboprint II	DM 89,–
RAM-Test für Amiga	DM 24,50
DPaint III	DM 248,–
Quarterback (Festplatten-Backup)	DM 119,–

Haben Sie Hard- oder Software für den Amiga entwickelt? Wir bieten Ihnen eine großzügige Umsatzprovision und eine ehrliche Abrechnung.
»»»Sprechen Sie uns an«««

Wir reparieren Ihren Amiga und Zubehör schnell und preisgünstig.

UPS-Versand: Nachnahme 10 DM, Vorauskasse 5 DM, Großgeräte nach UPS-Tabelle ohne Aufschlag, Ausland nur gegen Vorauskasse + 15DM

Fordern Sie unser kostenloses Info an
Händleranfragen erwünscht

HK-Computer
F. Hansmann & Th. Küpper GbR
Bonner Straße 37 ** 5000 Köln 1

Telefon: 0221/31 16 06 * Telefax: 0221/32 11 66
Mo-Fr. 10.00-13.30 u. 14.30-18.30, Sa. 10.00-14.00
Stadtsparkasse Köln, Kto 6342133, BLZ 37050198

******** hima computertechnik ********

SONDERANGEBOTE

Laufwerke für Amiga

3 1/2"-Amigalaufwerk, extern, abschaltbar und mit durchgeführtem Bus, amigafarbenes Metallgehäuse	DM 219,–
3 1/2"-Amigalaufwerk, intern, für Amiga 2000/500	DM 179,–
5 1/4"-Amigalaufwerk, extern, abschaltbar und mit durchgeführtem Bus, 40/80 Track, amigafarbenes Metallgehäuse	DM 269,–

Speichererweiterungen

512-KB-Erweiterungen für den Amiga 500, intern,
- mit akkugepufferter Echtzeituhr
- aufgebaut mit 1 MegaBit-Chips
- bei Bedarf abschaltbar

1,2 MB — DM 189,–

Händleranfragen und Angebote erwünscht
hima computertechnik Velbert
Burgstr. 13 · 5620 Velbert 1 · Telef: 0 20 51/5 67 67

Im nächsten Heft

Drucker gehören nach wie vor zur meistgekauften Peripherie. Wir haben für Sie zwei neue Drucker getestet: den MPS 1224 Farbdrucker von Commodore und den NEC P2 Plus.

Viren Leider konnten wir unseren Viren- Artikel in diesem Heft nicht bringen, weil es kurz nach Redaktionsschluß neue Erkenntnisse gab. Im nächsten Heft finden Sie jedoch alles über Viren und ihre Bekämpfung.

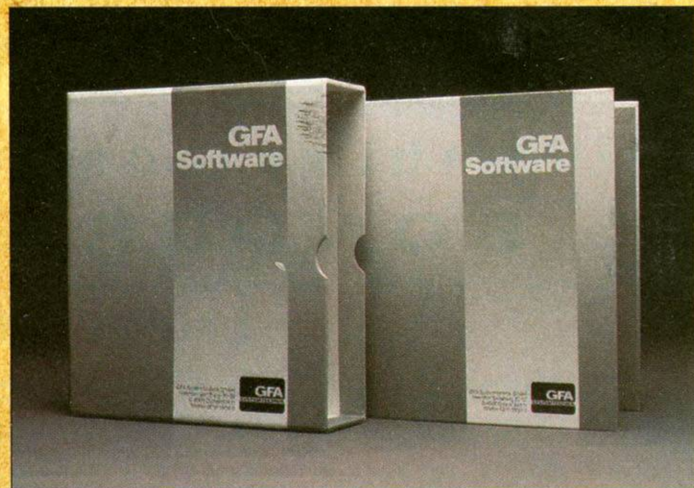


'Pinball Magic' heißt das neue Flipper-Spiel von Loriciel. Flippern Sie im nächsten Heft mit

Der GFA- Assembler ist da! Er soll demnächst seinen Platz neben DEVPAC, SEKA, A68K und anderen finden. Ob es ihm gelingen wird, haben wir für Sie getestet.

Spiele Flipper- Spiele auf dem Amiga erfreuen sich großer Beliebtheit. Das neue 'Pinball Magic' von Loriciel soll hier den Geschmack des Käufers treffen. Wie, das lesen Sie in der neuen AMIGA DOS.

Umbauten Einen Umbausatz besonderer Art stellen wir Ihnen im nächsten Heft vor. Für alle, die das Aussehen ihres A 500 zu einem PC-ähnlichen Gerät verändern wollen, testeten wir einen Komplett-Umrüstsatz. Schauen Sie selbst, ob es sich lohnt.



Wir stellen Ihnen den neuen GFA-Assembler vor

Big Agnus ist in aller Munde. Inzwischen bieten mehrere Händler Komplettbausätze für Amigas 'älteren Semesters' an. Wie der Umbau vonstatten geht und was vom neuen Chip zu erwarten ist, zeigt unser Bericht.

LIVE So nennt sich ein neues Video-Effekt-Gerät, mit dem man umfangreiche Möglichkeiten zur Bildgestaltung hat. Wie? Das erfahren Sie in unserem Test.

Und sonst? Tips & Tricks, Kurse, Werkstatt – alles das finden Sie in der nächsten AMIGA DOS.

Die nächste

AMIGA DOS

finden Sie ab

14. 03. '90

bei Ihrem Zeitschriftenhändler



Der verspätete Virenbeitrag, jetzt mit allen Informationen

Amiga Fraktal Generator 3D

Supergrafik im Sekundentakt

Vergessen Sie alles, was Sie bisher über Fraktalgrafik-Programme gehört haben – die unendliche Weite phantastischer Bilder erschließt sich nur über ein superschnelles Programm: **Fraktal Generator 3D**

High-Speed
Nur noch 7 Sekunden für das Urbild!

Super-Parallel-Projektion
Frei wählbarer horizontaler Blickwinkel mit 360 Grad:
Betrachten Sie das "Fraktalobjekt" von allen Seiten
Stufenloser vertikaler Blickwinkel:
Wahlweise Sicht von oben und unten, schräg oder in der Totalen

Speicherung im IFF-Standard
Einladen der Fraktal-Bilder in Mal- und Zeichenprogramme:
Verwendung als Hintergrund, Motiv oder Vorlage

Voller Bedienungskomfort
Auswahl komplett mit Pull-Down-Menüs
Wahlweise Maus- oder Tastensteuerung

Phantastische Farbmöglichkeiten
32 Farben im Low-Resolution-PAL-Modus
Eigenes Farbrequester mit stufenloser Schiebereglung

Mehrere separate Bildspeicher
Bis zu vier Bilder gleichzeitig abrufbar
Separate Farbzuordnung und Animationsmöglichkeit

Farb-Animationen
Phantastische Effekte durch Amiga-Color-Cycling

Amiga Fraktal Generator 3D

3 1/2"-Disk. Best.-Nr. 2901

69,- DM (unverbindliche Preisempfehlung)
Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

Inland:		**Ausland:**	
Einzelpreis	69,– DM	Einzelpreis	69,– DM
zzgl. Versandkosten	4,– DM	zzgl. Versandkosten	6,– DM
Endpreis	**73,– DM**	**Endpreis**	**75,– DM**

– Bitte benutzen Sie die Bestellkarte im Heft. –

DMV-Verlag · Postfach 250 · 3440 Eschwege



Strategy & Fantasy!

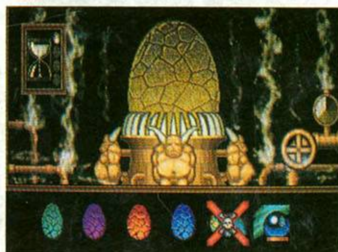
DRAGONS BREATH

Ein Fantasy-Trip, der das Beste aus den Elementen Arcade, Handelssimulation, Strategie und Magie in sich vereint. Finden Sie heraus, mit welchen Zaubersprüchen eine verfahrene strategische Situation noch zu Ihren Gunsten gewendet werden kann. Das große Buch der Zaubersprüche ist nicht vollständig! Es liegt an Ihnen, alle Möglichkeiten zu entdecken!

Für 1 – 3 Spieler auf Amiga und ST







Ariola Soft
Das Programm